



## MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

---

The MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature lets you configure your carrier-supporting-carrier network to enable Border Gateway Protocol (BGP) to transport routes and Multiprotocol Label Switching (MPLS) labels between the backbone carrier provider edge (PE) routers and the customer carrier customer edge (CE) routers using multiple paths. Previously, you had to use Label Distribution Protocol (LDP) to carry the labels and an Internal Gateway Protocol (IGP) to carry the routes between PE and CE routers to achieve the same goal.

The benefits of using BGP to distribute IPv4 routes and MPLS label routes are:

- BGP takes the place of an IGP and LDP in a Virtual Private Network (VPN) forwarding/routing instance (VRF) table. You can use BGP to distribute routes and MPLS labels. Using a single protocol instead of two simplifies the configuration and troubleshooting.
- BGP is the preferred routing protocol for connecting two Internet service providers (ISPs), mainly because of its routing policies and ability to scale. ISPs commonly use BGP between two providers. This feature enables those ISPs to use BGP.

This feature is an extension of the Carrier Supporting Carrier feature, introduced in Release 12.0(14)ST, which was based on LDP.

## Feature History for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

Release	Modification
12.0(21)ST	This feature was introduced.
12.0(22)S	This feature was implemented on the Cisco 12000 series router (see <a href="#">Table 1</a> for the line cards supported) and integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	Support was added for the Cisco 12000 Series Eight-Port OC-3c/STM-1c ATM Line Card (8-Port OC-3 ATM) and the Cisco 12000 Series Three-Port Gigabit Ethernet Line Card (3-Port GbE).
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
12.0(24)S	Support was added for the Cisco 12000 Series One-Port 10-Gigabit Ethernet Line Card (1-Port 10-GbE) and the Cisco 12000 Series Modular Gigabit Ethernet/Fast Ethernet Line Card (Modular GbE/FE) and implemented on Cisco IOS 12.0(24)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers.
12.0(27)S	Support was added for EBGp multipath on the provider edge (PE)-customer edge (CE) links.
12.0(29)S	Support was added for EBGp sessions between loopbacks of directly connected MPLS-enabled routers to provide for loadsharing between neighbors.
12.0(31)S	Support for the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature on customer-facing interfaces configured for the MPLS VPNs over IP Tunnels feature has been added to IP Services Engine (ISE) line cards on the Cisco 12000 Series Internet Router.

**Note**

Software images for Cisco 12000 series Internet routers have been deferred to Cisco IOS Release 12.0(27)S1.

**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 VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution, page 3](#)
- [Restrictions for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution, page 5](#)
- [Information About MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution, page 7](#)
- [How to Configure and Verify MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution, page 10](#)
- [Configuration Examples for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution, page 64](#)
- [Additional References, page 94](#)
- [Command Reference, page 96](#)
- [Glossary, page 98](#)

## Prerequisites for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

You should be able to configure MPLS VPNs with end-to-end (CE-to-CE router) pings working. To accomplish this, you need to know how to configure IGP routing protocols, LDP, and Multiprotocol Border Gateway Protocol (MP-BGP).

Make sure that the carrier supporting carrier provider edge (CSC-PE) routers and the carrier supporting carrier customer edge (CSC-CE) routers run images that support BGP label distribution. Otherwise, you cannot run external BGP (EBGP) between them.

[Table 1](#) lists the Cisco 12000 series line card support for Cisco IOS S releases.

**Table 1** Cisco I2000 Series Line Card Support for Cisco IOS S Releases

Type	Line Cards	Cisco IOS Release Supported
Packet Over SONET (POS)	4-Port OC-3 POS 8-Port OC-3 POS 16-Port OC-3 POS 1-Port OC-12 POS 4-Port OC-12 POS 1-Port OC-48 POS 4-Port OC-3 POS ISE 8-Port OC-3 POS ISE 16-Port OC-3 POS ISE 4-Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(22)S, 12.0(23)S, 12.0(27)S
Electrical Interface	6-Port DS3 12-Port DS3 6-Port E3 12-Port E3	12.0(22)S, 12.0(23)S, 12.0(27)S
Ethernet	3-Port GbE 1-Port 10-GbE Modular GbE/FE	12.0(23)S 12.0(24)S, 12.0(27)S

**Table 1 Cisco I2000 Series Line Card Support for Cisco IOS S Releases**

Type	Line Cards	Cisco IOS Release Supported
Asynchronous Transfer Mode (ATM)	4-Port OC-3 ATM 1-Port OC-12 ATM 4-Port OC-12 ATM 8-Port OC-3 ATM	12.0(22)S, 12.0(23)S, 12.0(27)S  12.0(23)S
Channelized Interface	2-Port CHOC-3 6-Port Ch T3 (DS1) 1-Port CHOC-12 (DS3) 1-Port CHOC-12 (OC-3) 4-Port CHOC-12 ISE 1-Port CHOC-48 ISE	12.0(22)S, 12.0(23)S, 12.0(27)S

**MPLS VPN Carrier Supporting Carrier over IP Tunnels**

Starting in IOS Release 12.0(31)S, you can configure the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature on supported IP Services Engine (ISE) line cards in a Cisco 12000 Series Internet Router, which have already been configured for the MPLS VPNs over IP Tunnels feature. The router must be deployed as a PE router in a service-provider core network.

The MPLS VPNs over IP Tunnels feature introduces the capability to deploy layer 3 VPN services over an IP core network using L2TPv3 multipoint tunnelling instead of MPLS. For more information, refer to [MPLS VPNs over IP Tunnels](#).

The MPLS VPN Carrier Supporting Carrier over IP Tunnels feature is supported only on customer-facing interfaces. The Cisco 12000 Series ISE line cards that support MPLS VPN Carrier Supporting Carrier over IP Tunnels are as follows:

- 4-port OC-3 POS ISE
- 8-port OC-3 POS ISE
- 16-port OC-3 POS ISE
- 4-port OC-12 POS ISE
- 1-port OC-48 POS ISE
- 1-port Channelized OC-12 (DS1) POS ISE
- 2.5G ISE SPA Interface Processor:
  - 2-Port Channelized T3 SPA
  - 4-Port Channelized T3 Serial SPA
- 4-port OC-12 ATM ISE
- 4-port OC3 ATM ISE
- 4-port GE ISE

# Restrictions for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

## Label Distribution

On a PE router, you can configure an interface for either BGP with labels or LDP. You cannot enable both types of label distribution on the same interface. If you switch from one protocol to the other, then you must disable the existing protocol on all interfaces before enabling the other protocol.

## Unsupported Features

This feature does not support the following:

- EBGp multihop between CSC-PE and CSC-CE routers
- EIBGP multipath load sharing

## Cisco Express Forwarding

The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding (CEF) or distributed Cisco Express Forwarding (dCEF) and MPLS.

## Supported Static Route Configurations

When configuring static routes in an MPLS or MPLS VPN environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco IOS releases that support the Tag Forwarding Information Base (TFIB), specifically Cisco IOS Releases 12.xT, 12.xM, and 12.0S. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), specifically Cisco IOS Release 12.2(25)S and later. Use the following guidelines when configuring static routes.

## Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in MPLS environment:

```
ip route destination-prefix mask interface next-hop-address
```

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

```
ip route destination-prefix mask interface1 next-hop1  
ip route destination-prefix mask interface2 next-hop2
```

## Unsupported Static Routes in an MPLS Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

```
ip route destination-prefix mask next-hop-address
```

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

```
ip route destination-prefix mask next-hop-address
```

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

```
ip route destination-prefix mask next-hop1
ip route destination-prefix mask next-hop2
```

Use the *interface* and *next-hop* arguments when specifying static routes.

#### Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface1 next-hop1*  
**ip route vrf** *vrf-name destination-prefix mask interface2 next-hop2*

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet Gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address global*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*  
(This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

```
ip route destination-prefix mask interface1 next-hop1
ip route destination-prefix mask interface2 next-hop2
```

#### Unsupported Static Routes in an MPLS VPN Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

```
ip route vrf destination-prefix mask next-hop-address global
```

The following **ip route** commands are not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

```
ip route vrf destination-prefix mask next-hop1 global
ip route vrf destination-prefix mask next-hop2 global
```

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

```
ip route vrf vrf-name destination-prefix mask next-hop1
ip route vrf vrf-name destination-prefix mask next-hop2
```

#### Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Router

The following **ip route vrf** command is supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table on the CE side. For example, the following command is supported when the destination-prefix is the CE router's loopback address, as in EBGp multihop cases.

```
ip route vrf vrf-name destination-prefix mask interface next-hop-address
```

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static non-recursive routes and a specific outbound interfaces:

```
ip route destination-prefix mask interface1 nexthop1  
ip route destination-prefix mask interface2 nexthop2
```

## Information About MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

To configure a carrier supporting carrier network that uses BGP to distribute routes and MPLS labels between the PE and CE routers of a backbone carrier and a customer carrier, you need to understand the following concepts:

- [Major Components of MPLS VPNs, page 8](#)
- [BGP Label Distribution, page 8](#)
- [Carrier Supporting Carrier Networks Supported for IPv4 BGP Label Distribution, page 9](#)

## Major Components of MPLS VPNs

An MPLS-based VPN network has three major components:

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.
- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers—MP-BGP propagates VRF reachability information to all members of a VPN community. MP-BGP peering needs to be configured in all PE routers within a VPN community.
- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

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

An MPLS VPN consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more CE routers attach to one or more PE routers. The PE routers use the MP-BGP to dynamically communicate with each other.

## BGP Label Distribution

This section contains the following topics:

- [BGP Routing Information, page 8](#)
- [Types of BGP Messages, page 8](#)
- [How BGP Sends MPLS Labels with Routes, page 9](#)

## BGP Routing Information

BGP routing information includes the following items:

- A network number (prefix)—The IP address of the destination.
- Autonomous system (AS) path—A list of other ASs through which a route passes on its way to the local router. The first AS in the list is the one closest to the local router; the last AS in the list is the one farthest from the local router, and it is usually the AS where the route began.
- Path attributes—Descriptors that provide other information about the AS path, for example, the next hop.

## Types of BGP Messages

MPLS labels are included in the update messages that a router sends. Routers exchange the following types of BGP messages:

- Open messages—After a router establishes a TCP connection with a neighboring router, the routers exchange open messages. This message contains the number of the AS to which the router belongs and the IP address of the router who sent the message.
- Update messages—When a router has a new, changed, or broken route, it sends an update message to the neighboring router. This message contains the Network Layer Reachability Information (NLRI), which lists the IP addresses of the usable routes. The update message includes any routes that are no longer usable. The update message also includes path attributes and the lengths of both



the usable and unusable paths. Labels for VPNv4 routes are encoded in the update message as specified in RFC 2858. The labels for the IPv4 routes are encoded in the update message as specified in RFC 3107.

- Keepalive messages—Routers exchange keepalive messages to determine if a neighboring router is still available to exchange routing information. The router sends these messages at regular intervals. (Sixty seconds is the default for Cisco routers.) The keepalive message does not contain routing data; it only contains a message header.
- Notification messages—When a router detects an error, it sends a notification message.

## How BGP Sends MPLS Labels with Routes

When BGP (both EBGP and IBGP) distributes a route, it can also distribute an MPLS label that is mapped to that route. The MPLS label mapping information for the route is carried in the BGP update message that contains the information about the route. If the next hop is not changed, the label is preserved.

When you issue the **neighbor send-label** command on both BGP routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

## Carrier Supporting Carrier Networks Supported for IPv4 BGP Label Distribution

This feature enables you to configure a carrier supporting carrier network that uses BGP to distribute routes and MPLS labels between the PE and CE routers of a backbone carrier and a customer carrier. The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be either of the following:

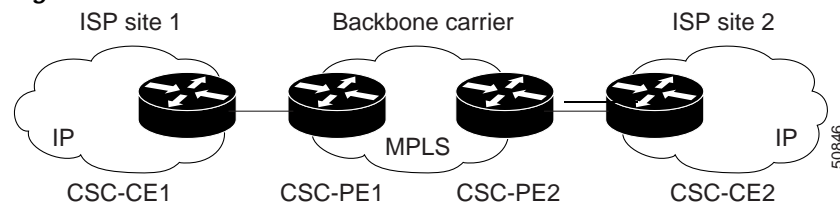
- [Customer Carrier Is an Internet Service Provider with an IP Core, page 9](#)
- [Customer Carrier Is an MPLS Service Provider With or Without VPN Services, page 10](#)

This document describes how to use BGP to distribute MPLS labels and routes for both types of customer carrier.

### Customer Carrier Is an Internet Service Provider with an IP Core

[Figure 1](#) shows a network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP.

**Figure 1** Network Where the Customer Carrier Is an ISP



In this configuration, the links between the CE and PE routers use EBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol IBGP to distribute VPNv4 routes.

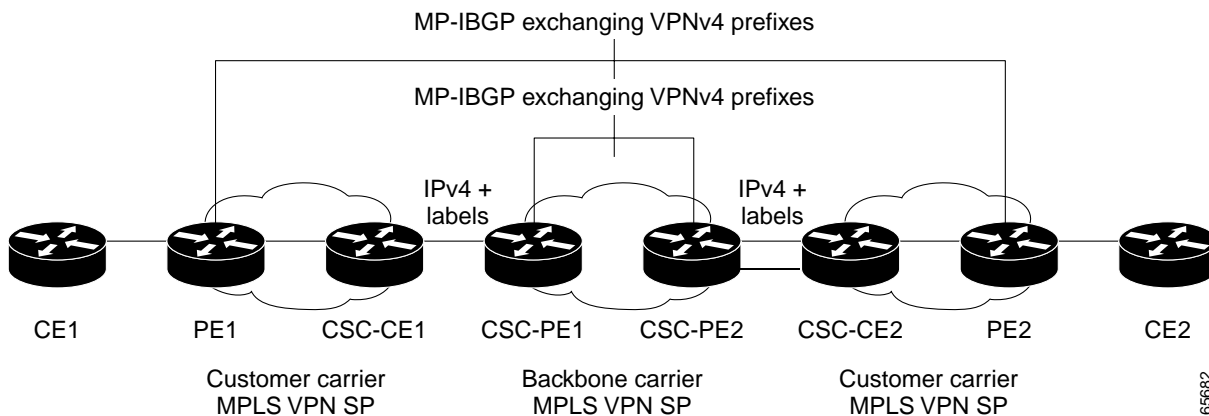
**Note**

If a router other than a Cisco router is used as a CSC-PE or CSC-CE, that router must support IPv4 BGP label distribution (RFC 3107). Otherwise, you cannot run EBGP with labels between the routers.

## Customer Carrier Is an MPLS Service Provider With or Without VPN Services

Figure 2 shows a network configuration where the backbone carrier and the customer carrier are BGP/MPLS VPN service providers. The customer carrier has two sites. Both the backbone carrier and the customer carrier use MPLS in their networks.

**Figure 2** Network Where the Customer Carrier Is an MPLS VPN Service Provider



In this configuration, the customer carrier can configure its network in one of the following ways:

- The customer carrier can run IGP and LDP in its core network. In this case, the CSC-CE1 router in the customer carrier redistributes the EBGP routes it learns from the CSC-PE1 router of the backbone carrier to IGP.
- The CSC-CE1 router of the customer carrier system can run an IPv4 and labels IBGP session with the PE1 router.

## How to Configure and Verify MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

This section contains the following tasks and processes that explain how to configure and verify the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature:

- [Identifying the Carrier Supporting Carrier Topology, page 11](#) (required)
- [Configuring and Verifying the Backbone Carrier Core, page 12](#) (required)
- [Configuring and Verifying the Links Between CSC-PE and CSC-CE Routers, page 18](#) (required)
- [Configuring and Verifying the Customer Carrier Network, page 52](#) (required)
- [Configuring and Verifying the Customer Site for Hierarchical VPNs, page 55](#) (required)



**Note** Configuration tasks are required. Verification tasks are optional.

## Identifying the Carrier Supporting Carrier Topology

Before you configure the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature, you need to identify both the backbone and customer carrier topology.

The main concern of this feature is the connectivity between the customer carrier and the backbone carrier. EBGP-based label distribution is configured on these links to enable MPLS between the customer and backbone carriers. MPLS VPNs described in the *MPLS Virtual Private Networks (VPNs)* feature module do not use MPLS on these carrier connections.

You need to identify the type of customer carrier as well as the topology of the customer carriers.



**Note** You can connect multiple CSC-CE routers to the same PE, or you can connect a single CSC-CE router to CSC-PEs using more than one interface to provide redundancy and multiple path support in CSC topology.

Perform this task to identify the carrier supporting carrier topology.

### SUMMARY STEPS

1. Identify the type of customer carrier, ISP or MPLS VPN service provider.
2. (For hierarchical VPNs only) Identify the CE routers.
3. (For hierarchical VPNs only) Identify the customer carrier core router configuration.
4. Identify the customer carrier edge (CSC-CE) routers.
5. Identify backbone carrier router configuration.

### DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the type of customer carrier, ISP or MPLS VPN service provider.	Sets up requirements for configuration of carrier supporting carrier with IPv4 and BGP label distribution. <ul style="list-style-type: none"> <li>• For an ISP, customer site configuration is not required.</li> <li>• For an MPLS VPN service provider, the customer site needs to be configured, as well as any task or step designated “for hierarchical VPNs only.”</li> </ul>
Step 2	(For hierarchical VPNs only) Identify the CE routers.	Sets up requirements for configuration of CE to PE connections.
Step 3	(For hierarchical VPNs only) Identify the customer carrier core router configuration.	Sets up requirements for connection configuration between core (P) routers and between P routers and edge routers (PE and CSC-CE routers).

	Command or Action	Purpose
Step 4	Identify the customer carrier edge (CSC-CE) routers.	Sets up requirements for configuration of CSC-CE to CSC-PE connections.
Step 5	Identify the backbone carrier router configuration.	Sets up requirements for connection configuration between core (CSC-Core) routers and between CSC-Core routers and edge routers (CSC-CE and CSC-PE routers).

### What to Do Next

Set up your carrier supporting carrier networks for the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature starting with the [“Configuring and Verifying the Backbone Carrier Core” section on page 12](#).

## Configuring and Verifying the Backbone Carrier Core

Configuring the backbone carrier core in an MPLS VPN carrier supporting carrier network with BGP label distribution requires setting up connectivity and routing functions for the CSC-Core and the CSC-PE routers.

### Prerequisites

Before you configure a backbone carrier core for the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature, you must configure the following on the CSC-Core routers:

- An IGP routing protocol—BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see the “IP Routing Protocols” chapter in the [Cisco IOS IP Configuration Guide, Release 12.0](#).
- Label Distribution Protocol (LDP). For information, see the [MPLS Label Distribution Protocol \(LDP\)](#).



#### Note

These prerequisites must be configured first.

Configuring and verifying the CSC-Core (backbone carrier) involves the following tasks:

- [Verifying IP Connectivity and LDP Configuration in the CSC-Core, page 12](#) (optional)
- [Configuring VRFs for CSC-PE Routers, page 14](#) (required)
- [Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier, page 16](#) (required)

## Verifying IP Connectivity and LDP Configuration in the CSC-Core

Perform this task to verify IP connectivity and LDP configuration in the CSC-Core.

### SUMMARY STEPS

1. **enable**
2. **ping** [protocol] {host-name | system-address}
3. **trace** [protocol] [destination]

4. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
5. **show mpls ldp discovery** [[**vrf** *vrf-name*] | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**]
9. **show ip route**
10. **disable**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p>
Step 2	<p><b>ping</b> [<i>protocol</i>] {<i>host-name</i>   <i>system address</i>}</p> <p><b>Example:</b> Router# ping ip &lt;CSC-Core-address&gt;</p>	<p>(Optional) Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks.</p> <ul style="list-style-type: none"> <li>Use the <b>ping ip</b> command to verify the connectivity from one CSC-Core router to another.</li> </ul>
Step 3	<p><b>trace</b> [<i>protocol</i>] [<i>destination</i>]</p> <p><b>Example:</b> Router# trace ip destination-address</p>	<p>(Optional) Discovers the routes that packets will actually take when traveling to their destination.</p> <ul style="list-style-type: none"> <li>Use the <b>trace</b> command to verify the path that a packet goes through before reaching the final destination. The <b>trace</b> command can help isolate a trouble spot if two routers cannot communicate.</li> </ul>
Step 4	<p><b>show mpls forwarding-table</b> [<b>vrf</b> <i>vrf-name</i>] [{<i>network</i> {<i>mask</i>   <i>length</i>}   <b>labels</b> <i>label</i> [- <i>label</i>]   <b>interface</b> <i>interface</i>   <b>next-hop</b> <i>address</i>   <b>lsp-tunnel</b> [<i>tunnel-id</i>]}] [<b>detail</b>]</p> <p><b>Example:</b> Router# show mpls forwarding-table</p>	<p>(Optional) Display the contents of the MPLS label forwarding information base (LFIB).</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls forwarding-table</b> command to verify that MPLS packets are being forwarded.</li> </ul>
Step 5	<p><b>show mpls ldp discovery</b> [[<b>vrf</b> <i>vrf-name</i>]   <b>all</b>]</p> <p><b>Example:</b> Router# show mpls ldp discovery</p>	<p>(Optional) Displays the status of the LDP discovery process.</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls ldp discovery</b> command to verify that LDP is operational in the CSC-Core.</li> </ul>
Step 6	<p><b>show mpls ldp neighbor</b> [[<b>vrf</b> <i>vrf-name</i>] [<i>address</i>   <i>interface</i>] [<b>detail</b>]   <b>all</b>]</p> <p><b>Example:</b> Router# show mpls ldp neighbor</p>	<p>(Optional) Displays the status of LDP sessions.</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls ldp neighbor</b> command to verify LDP configuration in the CSC-Core.</li> </ul>

	Command or Action	Purpose
Step 7	<pre>show ip cef [vrf vrf-name] [network [mask]] [longer-prefixes] [detail]</pre> <p><b>Example:</b> Router# show ip cef</p>	(Optional) Displays entries in the forwarding information base (FIB). <ul style="list-style-type: none"> <li>Use the <b>show ip cef</b> command to check the forwarding table (prefixes, next hops, and interfaces).</li> </ul>
Step 8	<pre>show mpls interfaces [[vrf vrf-name] [interface] [detail]   [all]]</pre> <p><b>Example:</b> Router# show mpls interfaces</p>	(Optional) Displays information about one or more or all interfaces that are configured for label switching. <ul style="list-style-type: none"> <li>Use the <b>show mpls interfaces</b> command to verify that the interfaces are configured to use LDP.</li> </ul>
Step 9	<pre>show ip route</pre> <p><b>Example:</b> Router# show ip route</p>	(Optional) Displays IP routing table entries. <ul style="list-style-type: none"> <li>Use the <b>show ip route</b> command to display the entire routing table, including host IP address, next hop, interface, and so forth.</li> </ul>
Step 10	<pre>disable</pre> <p><b>Example:</b> Router# disable</p>	(Optional) Returns to user mode.

### Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

### Additional Information

For a configuration example for this task, see the [“Verifying IP Connectivity and LDP Configuration in the CSC-Core: Example”](#) section on page 65.

## Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN forwarding/routing instances (VRFs) for the backbone carrier edge (CSC-PE) routers.

### SUMMARY STEPS

- enable**
- configure terminal**
- ip vrf vrf-name**
- rd route-distinguisher**
- route-target {import | export | both} route-target-ext-community**
- import map route-map**
- exit**
- interface type number**

9. **ip vrf forwarding** *vrf-name*
10. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>ip vrf vrf-name</pre> <p><b>Example:</b> Router(config)# ip vrf vpn1 </p>	<p>Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
Step 4	<pre>rd route-distinguisher</pre> <p><b>Example:</b> Router(config-vrf)# rd 100:1 </p>	<p>Creates routing and forwarding tables.</p> <ul style="list-style-type: none"> <li>The <i>route-distinguisher</i> argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: <ul style="list-style-type: none"> <li>16-bit AS number: your 32-bit number, for example, 101:3</li> <li>32-bit IP address: your 16-bit number, for example, 192.168.122.15:1</li> </ul> </li> </ul>
Step 5	<pre>route-target {import  export   both} route-target-ext-community</pre> <p><b>Example:</b> Router(config-vrf)# route-target import 100:1 </p>	<p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> <li>The <b>import</b> keyword imports routing information from the target VPN extended community.</li> <li>The <b>export</b> keyword exports routing information to the target VPN extended community.</li> <li>The <b>both</b> keyword imports routing information from and exports routing information to the target VPN extended community.</li> <li>The <i>route-target-ext-community</i> 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.</li> </ul>
Step 6	<pre>import map route-map</pre> <p><b>Example:</b> Router(config-vrf)# import map vpn1-route-map </p>	<p>(Optional) Configures an import route map for a VRF.</p> <ul style="list-style-type: none"> <li>The <i>route-map</i> argument specifies the route map to be used as an import route map for the VRF.</li> </ul>

	Command or Action	Purpose
Step 7	<code>exit</code>  <b>Example:</b> <code>Router(config-vrf)# exit</code>	(Optional) Exits to global configuration mode.
Step 8	<code>interface type number</code>  <b>Example:</b> <code>Router(config)# interface Ethernet5/0</code>	Specifies the interface to configure. <ul style="list-style-type: none"> <li>The <i>type</i> argument specifies the type of interface to be configured.</li> <li>The <i>number</i> argument specifies the port, connector, or interface card number.</li> </ul>
Step 9	<code>ip vrf forwarding vrf-name</code>  <b>Example:</b> <code>Router(config-if)# ip vrf forwarding vpn1</code>	Associates a VRF with the specified interface or subinterface. <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
Step 10	<code>end</code>  <code>Router(config-if)# end</code>	(Optional) 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.

### Additional Information

For a configuration example for this task, see the [“Configuring VRFs for CSC-PE Routers: Example” section on page 67](#).

## Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

### SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *as-number*
- no bgp default ipv4-unicast**
- neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
- neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type*
- address-family vpnv4** [**unicast**]
- neighbor** {*ip-address* | *peer-group-name*} **send-community** **extended**
- neighbor** {*ip-address* | *peer-group-name*} **activate**
- end**



## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>router bgp as-number</pre> <p><b>Example:</b> Router(config)# router bgp 100 </p>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>
Step 4	<pre>no bgp default ipv4-unicast</pre> <p><b>Example:</b> Router(config-router)# no bgp default ipv4-unicast </p>	<p>(Optional) Disables the IPv4 unicast address family on all neighbors.</p> <ul style="list-style-type: none"> <li>Use the <b>no</b> form of the <b>bgp default-unicast</b> command if you are using this neighbor for MPLS routes only.</li> </ul>
Step 5	<pre>neighbor {ip-address   peer-group-name} remote-as as-number</pre> <p><b>Example:</b> Router(config-router)# neighbor &lt;CSC-Core-ip-address&gt; remote-as 100 </p>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>
Step 6	<pre>neighbor {ip-address   peer-group-name} update-source interface-type</pre> <p><b>Example:</b> Router(config-router)# neighbor &lt;CSC-Core-ip-address&gt; update-source loopback0 </p>	<p>Allows BGP sessions to use a specific operational interface for TCP connections.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>interface-type</i> argument specifies the interface to be used as the source.</li> </ul>
Step 7	<pre>address-family vpnv4 [unicast]</pre> <p><b>Example:</b> Router(config-router)# address-family vpnv4 </p>	<p>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.</p> <ul style="list-style-type: none"> <li>The optional <b>unicast</b> keyword specifies VPNv4 unicast address prefixes.</li> </ul>

	Command or Action	Purpose
Step 8	<pre>neighbor {ip-address   peer-group-name} send-community extended</pre> <p><b>Example:</b>  Router(config-router-af)# neighbor pp.0.0.1 send-community extended</p>	<p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>
Step 9	<pre>neighbor {ip-address   peer-group-name} activate</pre> <p><b>Example:</b>  Router(config-router-af)# neighbor &lt;CSC-Core-ip-address&gt; activate</p>	<p>Enables the exchange of information with a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>
Step 10	<pre>end</pre> <p><b>Example:</b>  Router(config-router-af)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

### 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.

### Additional Information

For a configuration example for this task, see the [“Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier: Example”](#) section on page 67.

## Configuring and Verifying the Links Between CSC-PE and CSC-CE Routers

Perform the following tasks to configure and verify links between a carrier supporting carrier backbone edge (CSC-PE) router and the carrier supporting carrier customer edge (CSC-CE) router.

- [Configuring Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers, page 19](#) (optional)
- [Configuring Peering of the Loopback Interface of Directly Connected CSC-PE and CSC-PE Routers, page 28](#) (optional)
- [Configuring Route Maps on the CSC-PE Routers, page 45](#) (optional)

## Configuring Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers

Configuring and verifying the links between the carrier supporting carrier backbone edge (CSC-PE) router and the carrier supporting carrier customer edge (CSC-CE) router involves the following tasks:

- [Configuring EBGP with send-label Option for Carrier Supporting Carrier on CSC-PE Routers, page 19](#) (required)
- [Configuring EBGP with send-label Option for Carrier Supporting Carrier on CSC-CE Routers, page 21](#) (required)
- [Verifying Labels in the CSC-PE Routers, page 24](#) (optional)
- [Verifying Labels in the CSC-CE Routers, page 26](#) (optional)

Figure 4 shows the configuration for the peering with directly connected interfaces between CSC-PE and CSC-CE routers. This configuration is used as the example in the tasks that follow.

**Figure 3** Configuration for Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers



### Note

When External Border Gateway Protocol (EBGP) sessions come up, BGP automatically generates the **mpls bgp forwarding** command on the connecting interface.

## Configuring EBGP with send-label Option for Carrier Supporting Carrier on CSC-PE Routers

Perform this task to configure Carrier Supporting Carrier on the CSC-PE routers so that they can distribute BGP routes with MPLS labels.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf vrf-name**]
5. **maximum paths** *number-paths* (optional for EBGP multipath between the CSC-PE and CSC-CEs)
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **as-override**
9. **neighbor** *ip-address* **send-label**
10. **exit-address-family**
11. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<pre>router bgp as-number</pre> <p><b>Example:</b> Router(config)# router bgp 100</p>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>
Step 4	<pre>address-family ipv4 [multicast   unicast   vrf vrf-name]</pre> <p><b>Example:</b> Router(config-router)# address-family ipv4 vrf vpn1</p>	<p>Specifies the IPv4 address family type and enters address family configuration mode.</p> <ul style="list-style-type: none"> <li>The <b>multicast</b> keyword specifies IPv4 multicast address prefixes.</li> <li>The <b>unicast</b> keyword specifies IPv4 unicast address prefixes.</li> <li>The <b>vrf vrf-name</b> keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</li> </ul>
Step 5	<pre>maximum-paths number-paths</pre> <p><b>Example:</b> Router(config-router-af)# maximum-paths 2</p>	<p>(Optional) Controls the maximum number of parallel routes an IP routing protocol can support.</p> <ul style="list-style-type: none"> <li>The <i>number-paths</i> argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table, in the range from 1 to 6.</li> </ul>
Step 6	<pre>neighbor {ip-address   peer-group-name} remote-as as-number</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 remote-as 200</p>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>

	Command or Action	Purpose
Step 7	<pre>neighbor {ip-address   peer-group-name} activate</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 activate</p>	<p>Enables the exchange of information with a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>
Step 8	<pre>neighbor ip-address as-override</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 as-override</p>	<p>Configures a PE router to override the ASN of a site with the ASN of a provider.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the router that is to be overridden with the ASN provided.</li> </ul>
Step 9	<pre>neighbor ip-address send-label</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 send-label</p>	<p>Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighboring router.</li> </ul>
Step 10	<pre>exit-address-family</pre> <p><b>Example:</b> Router(config-router-af)# exit-address-family</p>	<p>Exits address family configuration mode.</p>
Step 11	<pre>end</pre> <p><b>Example:</b> Router(config-router)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

## Troubleshooting Tips

Enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. Make sure you see the following line in the command output under Neighbor capabilities:

```
IPv4 MPLS Label capability:advertised and received
```

## Additional Information

For a configuration example for this task, see the [“Configuring EBGP Link with send-label Option for Carrier Supporting Carrier on CSC-PE Routers: Examples”](#) section on page 68.

## Configuring EBGP with send-label Option for Carrier Supporting Carrier on CSC-CE Routers

Perform this task to configure Carrier Supporting Carrier on the CSC-CE routers so that they can distribute BGP routes with MPLS labels.

## SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *as-number*
- maximum paths** *number-paths* (optional for EBGP multipath between the CSC-PE and CSC-CEs)

5. **address-family ipv4** [**multicast** | **unicast** | **vrf vrf-name**]
6. **redistribute protocol**
7. **neighbor** {*ip-address* | *peer-group-name*} **remote-as as-number**
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **neighbor ip-address send-label**
10. **exit-address-family**
11. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>router bgp as-number</b>  <b>Example:</b> Router(config)# router bgp 200	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>
Step 4	<b>maximum-paths number-paths</b>  <b>Example:</b> Router(config-router)# maximum-paths 2	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. <ul style="list-style-type: none"> <li>The <i>number-paths</i> argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table, in the range from 1 to 6.</li> </ul>
Step 5	<b>address-family ipv4</b> [ <b>multicast</b>   <b>unicast</b>   <b>vrf vrf-name</b> ]  <b>Example:</b> Router(config-router)# address-family ipv4	Specifies the IPv4 address family type and enters address family configuration mode. <ul style="list-style-type: none"> <li>The <b>multicast</b> keyword specifies IPv4 multicast address prefixes.</li> <li>The <b>unicast</b> keyword specifies IPv4 unicast address prefixes.</li> <li>The <b>vrf vrf-name</b> keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</li> </ul>

	Command or Action	Purpose
Step 6	<pre>redistribute protocol</pre> <p><b>Example:</b> Router(config-router-af)# redistribute static</p>	<p>Redistributes routes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> <li>The <i>protocol</i> argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: <b>bgp</b>, <b>egp</b>, <b>igrp</b>, <b>isis</b>, <b>ospf</b>, <b>mobile</b>, <b>static [ip]</b>, <b>connected</b>, and <b>rip</b>. <ul style="list-style-type: none"> <li>The <b>static [ip]</b> keyword redistributes IP static routes. The optional <b>ip</b> keyword is used when you redistribute static routes into IS-IS.</li> <li>The <b>connected</b> keyword refers to routes which are established automatically when IP is enabled on an interface. For routing protocols such as OSPF and IS-IS, these routes are redistributed as external to the autonomous system.</li> </ul> </li> </ul>
Step 7	<pre>neighbor {ip-address   peer-group-name} remote-as as-number</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.2 remote-as 100</p>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>
Step 8	<pre>neighbor {ip-address   peer-group-name} activate</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.2 activate</p>	<p>Enables the exchange of information with a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>
Step 9	<pre>neighbor ip-address send-label</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.2 send-label</p>	<p>Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighboring router.</li> </ul>
Step 10	<pre>exit-address-family</pre> <p><b>Example:</b> Router(config-router-af)# exit-address-family</p>	<p>Exits from the address family configuration mode.</p>
Step 11	<pre>end</pre> <p><b>Example:</b> Router(config-router)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

## Additional Information

For a configuration example for this task, see the [“Configuring EBGP Link with send-label Option for Carrier Supporting Carrier on CSC-CE Routers: Examples”](#) section on page 70.

## Verifying Labels in the CSC-PE Routers

Perform this task to verify the labels in the CSC-PE routers.

### SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {all | rd *route-distinguisher* | vrf *vrf-name*} [summary] [labels]
3. **show mpls interfaces** [all]
4. **show ip route vrf** *vrf-name* [*prefix*]
5. **show ip bgp vpnv4** {all | rd *route-distinguisher* | vrf *vrf-name*} [summary] [labels]
6. **show ip cef** [vrf *vrf-name*] [*network* [*mask*]] [longer-prefixes] [detail]
7. **show mpls forwarding-table** [vrf *vrf-name*] [{*network* {*mask* | *length*} | labels *label* [- *label*] | interface *interface* | next-hop *address* | lsp-tunnel [*tunnel-id*]}] [detail]
8. **traceroute VRF** [*vrf-name*] *ip-address*
9. **disable**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>show ip bgp vpnv4</b> {all   rd <i>route-distinguisher</i>   vrf <i>vrf-name</i> } [summary] [labels]  <b>Example:</b> Router# show ip bgp vpnv4 all summary	(Optional) Displays VPN address information from the BGP table. <ul style="list-style-type: none"> <li>• Use the <b>show ip bgp vpnv4 all summary</b> command to check that the BGP session is up and running between the CSC-PE routers and the CSC-CE routers. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.</li> </ul>
Step 3	<b>show mpls interfaces</b> [all]  <b>Example:</b> Router# show mpls interfaces all	(Optional) Displays information about one or more interfaces that have been configured for label switching. <ul style="list-style-type: none"> <li>• Use the <b>show mpls interfaces all</b> command to check that MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. Check that LDP is turned off on the VRF because EBGP distributes the labels.</li> </ul>



Command or Action	Purpose
<p><b>Step 4</b></p> <pre>show ip route vrf vrf-name [prefix]</pre> <p><b>Example:</b> Router# show ip route vrf vpn1 &lt;PE-prefix&gt;</p>	<p>(Optional) Displays the IP routing table associated with a VRF.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip route vrf</b> command to check that the prefixes for the PE routers are in the routing table of the CSC-PE routers.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.</p>
<p><b>Step 5</b></p> <pre>show ip bgp vpnv4 {all   rd route-distinguisher   vrf vrf-name} [summary] [labels]</pre> <p><b>Example:</b> Router# show ip bgp vpnv4 vrf vpn1 labels</p>	<p>(Optional) Displays VPN address information from the BGP table.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip bgp vpnv4 vrf vrf-name labels</b> command to check that the prefixes for the customer carrier MPLS service provider networks are in the BGP table and have the appropriate labels.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.</p>
<p><b>Step 6</b></p> <pre>show ip cef [vrf vrf-name] [network [mask]] [longer-prefixes] [detail]</pre> <p><b>Example:</b> Router# show ip cef vrf vpn1 &lt;PE-prefix&gt;</p> <p>Router# show ip cef vrf vpn1 &lt;PE-prefix&gt; detail</p>	<p>(Optional) Displays entries in the forwarding information base (FIB) or displays a summary of the FIB.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip cef vrf</b> and the <b>show ip cef vrf detail</b> commands to check that the prefixes of the PE routers are in the CEF table.</li> </ul>
<p><b>Step 7</b></p> <pre>show mpls forwarding-table [vrf vrf-name] [{network {mask   length}   labels label [- label]   interface interface   next-hop address   lsp-tunnel [tunnel-id]}] [detail]</pre> <p><b>Example:</b> Router# show mpls forwarding-table vrf vpn1 &lt;PE-prefix&gt;</p> <p>Router# show mpls forwarding-table vrf vpn1 &lt;PE-prefix&gt; detail</p>	<p>(Optional) Displays the contents of the MPLS forwarding information base (LFIB).</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls forwarding-table</b> command with the <b>vrf</b> keyword and both the <b>vrf</b> and <b>detail</b> keywords to check that the prefixes for the PE routers in the local customer MPLS VPN service provider are in the LFIB.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.</p>

Command or Action	Purpose
<p><b>Step 8</b></p> <pre>traceroute VRF [vrf-name] ip-address</pre> <p><b>Example:</b> Router# traceroute vrf vpn2 jj.jj.jj.jj</p>	<p>Shows the routes that packets follow traveling through a network to their destination.</p> <ul style="list-style-type: none"> <li>Use the <b>traceroute VRF</b> command to check the data path and transport labels from a PE to a destination CE router.</li> </ul> <p><b>Note</b> This command works with MPLS-aware traceroute only if the backbone routers are configured to propagate and generate IP Time to Live (TTL) information. For more information, see the documentation on the <b>mpls ip propagate-ttl</b> command.</p> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.</p>
<p><b>Step 9</b></p> <pre>disable</pre> <p><b>Example:</b> Router# disable</p>	<p>(Optional) Exits to user EXEC mode.</p>

### Additional Information

For a configuration example for this task, see the [“Verifying Labels in the CSC-PE Routers: Examples” section on page 72](#).

### Verifying Labels in the CSC-CE Routers

Perform this task to verify the labels in the CSC-CE routers.

### SUMMARY STEPS

- enable**
- show ip bgp summary**
- show ip route [address]**
- show mpls ldp bindings [network {mask | length}]**
- show ip cef [network [mask]] [longer-prefixes] [detail]**
- show mpls forwarding-table [vrf vrf-name] [{network {mask | length} | labels label [- label] | interface interface | next-hop address | lsp-tunnel [tunnel-id]}] [detail]**
- show ip bgp labels**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p>
Step 2	<pre>show ip bgp summary</pre> <p><b>Example:</b> Router# show ip bgp summary </p>	<p>(Optional) Displays the status of all BGP connections.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip bgp summary</b> command to check that the BGP session is up and running on the CSC-CE routers.</li> </ul>
Step 3	<pre>show ip route [address]</pre> <p><b>Example:</b> Router# show ip route PE-address </p>	<p>(Optional) Displays IP routing table entries.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip route</b> command to check that the loopback address of the local and remote PE routers are in the routing table.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.</p>
Step 4	<pre>show mpls ldp bindings [network {mask   length}]</pre> <p><b>Example:</b> Router# show mpls ldp bindings PE-prefix 255.255.255.255 </p>	<p>(Optional) Displays the contents of the label information base (LIB).</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls ldp bindings</b> command to check that the prefix of the local PE router is in the MPLS LDP bindings.</li> </ul>
Step 5	<pre>show ip cef [network [mask]] [longer-prefixes] [detail]</pre> <p><b>Example:</b> Router# show ip cef &lt;PE-prefix&gt;  Router# show ip cef &lt;PE-prefix&gt; detail </p>	<p>(Optional) Displays entries in the forwarding information base (FIB) or a summary of the FIB.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip cef</b> and the <b>show ip cef detail</b> commands to check that the prefixes of the local and remote PE routers are in the CEF table.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.</p>

Command or Action	Purpose
<p><b>Step 6</b></p> <pre>show mpls forwarding-table [vrf vrf-name] [ {network {mask   length}   labels label [- label]   interface interface   next-hop address   lsp-tunnel [tunnel-id]}] [detail]</pre> <p><b>Example:</b></p> <pre>Router# show mpls forwarding-table &lt;PE-prefix&gt;</pre> <pre>Router# show mpls forwarding-table &lt;PE-prefix&gt; detail</pre>	<p>(Optional) Displays the contents of the MPLS LFIB.</p> <ul style="list-style-type: none"> <li>Use the <b>show mpls forwarding-table</b> and <b>show mpls forwarding-table detail</b> commands to check that the prefixes of the local and remote PE routers are in the MPLS forwarding table.</li> </ul> <p><b>Note</b> If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.</p>
<p><b>Step 7</b></p> <pre>show ip bgp labels</pre> <p><b>Example:</b></p> <pre>Router# show ip bgp labels</pre>	<p>(Optional) Displays information about MPLS labels from the EBGp route table.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip bgp labels</b> command to check that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks.</li> </ul>

### Additional Information

For a configuration example for this task, see the [“Verifying Labels in the CSC-CE Routers: Examples” section on page 78](#).

## Configuring Peering of the Loopback Interface of Directly Connected CSC-PE and CSC-PE Routers

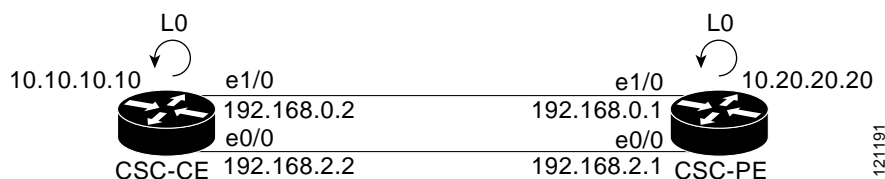
This functionality is provided with the release of the Cisco IOS feature MPLS VPN—Loadbalancing Support for Inter-AS and CSC VPNs.

This section describes the tasks you need to do to configure peering of loopback interfaces of directly connected CSC-PE and CSC-CE routers. The tasks include the following:

- [Configuring Loopback Interface Addresses for CSC-PE and CSC-CE Routers, page 29](#) (required)
- [Configuring /32 Static Routes to the EBGp Neighbor Loopback, page 31](#) (required)
- [Configuring Forwarding on Connecting Loopback Interfaces, page 35](#) (required)
- [Configuring an EBGp Session Between the Loopbacks, page 38](#) (required)
- [Verifying That Load Balancing Occurs Between Loopbacks, page 44](#) (optional)

[Figure 4](#) shows the loopback configuration for directly connected CSC-PE and CSC-CE routers. This configuration is used as the example in the tasks that follow.

**Figure 4** Loopback Interface Configuration for Directly Connected CSC-PE and CSC-CE Routers



## Restrictions for the MPLS VPN—Loadbalancing Support for Inter-AS and CSC VPNs Feature

The MPLS VPN—Loadbalancing Support for Inter-AS and CSC VPNs feature does *not* apply to the following:

- Load balancing using the Label Distribution Protocol (LDP) plus an Internal Gateway Protocol (IGP) between CSC-PE routers and CSC-CE routers

## Configuring Loopback Interface Addresses for CSC-PE and CSC-CE Routers

Perform the following task to configure loopback addresses. Loopback addresses need to be configured for both the CSC-PE and the CSC-CE router.



### Note

Configuration of a loopback interface address on the CSC-PE router requires the enabling of a Virtual Private Network (VPN) routing/forwarding instance (VRF). The CSC-CE router loopback interface does not require the enabling of VRF.

### Configuring Loopback Interface Addresses for CSC-PE Routers

Perform this task to configure loopback interface addresses for CSC-PE routers.

## SUMMARY STEPS—CSC-PE

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **end**

## DETAILED STEPS—CSC-PE

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>interface loopback</b> <i>interface-number</i>  <b>Example:</b> Router(config)# interface loopback0	Configures a software-only virtual interface that emulates an interface that is always up.  • The <i>interface-number</i> argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.

	Command or Action	Purpose
Step 4	<code>ip vrf forwarding vrf-name</code>  <b>Example:</b> Router(config-if)# ip vrf forwarding vpn1	Associates a VRF with the specified interface or subinterface. <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
Step 5	<code>ip address ip-address mask [secondary]</code>  <b>Example:</b> Router(config-if)# ip address 10.20.20.20 255.255.255.255	Sets a primary or secondary IP address for an interface. <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address.</li> <li>The <i>mask</i> argument is the mask for the associated IP subnet.</li> <li>The <b>secondary</b> keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.</li> </ul>
Step 6	<code>end</code>  <b>Example:</b> Router(config)# end	Exits to privileged EXEC mode.

### Examples—CSC-PE

The following example shows the configuration of a loopback address for the CSC-PE router:

```
configure terminal
interface loopback0
    ip vrf forwarding
    ip address 10.20.20.20 255.255.255.255.255
```

### Configuring Loopback Interface Addresses for CSC-CE Routers

Perform this task to configure loopback interface addresses for CSC-CE routers.

### SUMMARY STEPS—CSC-CE

- enable**
- configure terminal**
- interface loopback** *interface-number*
- ip address** *ip-address mask [secondary]*
- end**

## DETAILED STEPS—CSC-CE

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>Enter your password if prompted.</li></ul>
Step 2	<code>configure terminal</code>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<code>interface loopback interface-number</code>  <b>Example:</b> Router(config)# interface loopback0	Configures a software-only virtual interface that emulates an interface that is always up. <ul style="list-style-type: none"><li>The <i>interface-number</i> argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.</li></ul>
Step 4	<code>ip address ip-address mask [secondary]</code>  <b>Example:</b> Router(config-if)# ip address 10.10.10.10 255.255.255.255	Sets a primary or secondary IP address for an interface. <ul style="list-style-type: none"><li>The <i>ip-address</i> argument is the IP address.</li><li>The <i>mask</i> argument is the mask for the associated IP subnet.</li><li>The <b>secondary</b> keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.</li></ul>
Step 5	<code>end</code>  <b>Example:</b> Router(config)# end	Exits to privileged EXEC mode.

## Examples—CSC-CE

The following example shows the configuration of a loopback address for the CSC-CE router:

```
configure terminal
interface loopback0
    ip address 10.10.10.10 255.255.255.255
```

## Configuring /32 Static Routes to the EBGp Neighbor Loopback

Perform the following task to configure /32 static routes to the EBGp neighbor lookback.

A /32 static route is established with the following commands:

```
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet1/0 Y.Y.Y.Y
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet0/0 Z.Z.Z.Z
```

Where *X.X.X.X* is the neighboring loopback address and Ethernet 1/0 and Ethernet 0/0 are the links connecting the peering routers. *Y.Y.Y.Y* and *Z.Z.Z.Z* are the respective next-hop addresses on the interfaces.

**Configuring /32 Static Routes to the EBGW Neighbor Loopback for the CSC-PE Router**

Perform the following task to configure /32 static routes to the EBGW neighbor loopback for the CSC-PE router.

**SUMMARY STEPS—CSC-PE**

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name prefix mask {ip-address | interface-type interface-number [ip-address]}* [global] [distance] [name] [permanent] [tag tag]
4. **end**

**DETAILED STEPS—CSC-PE**

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



	Command or Action	Purpose
Step 3	<pre>ip route vrf vrf-name prefix mask {ip-address   interface-type interface-number [ip-address]} [global] [distance] [name] [permanent] [tag tag]</pre> <p><b>Example:</b></p> <pre>Router(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet1/0 168.192.0.2</pre> <pre>Router(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet0/0 168.192.2.2</pre>	<p>Establishes static routes for a Virtual private Network (VPN) routing/ forwarding instance (VRF).</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name of the VRF for the static route.</li> <li>The <i>prefix</i> argument is the IP route prefix for the destination.</li> <li>The <i>mask</i> argument is the prefix mask for the destination.</li> <li>The <i>ip-address</i> argument is the IP address of the next hop that you can use to reach the destination network.</li> <li>The <i>interface-type</i> and <i>interface-number</i> arguments are the network interface type and interface number.</li> <li>The <b>global</b> keyword specifies that the given next hop address is in the non-VRF routing table.</li> <li>The <i>distance</i> argument is an administrative distance.</li> <li>The <i>name</i> argument applies a name to the specified route.</li> <li>The <b>permanent</b> keyword specifies that the route is not to be removed, even if the interface shuts down.</li> <li>The <b>tag tag</b> keyword-argument pair names a tag value that can be used as a “match” value for controlling redistribution via route maps.</li> </ul>
Step 4	<pre>end</pre> <p><b>Example:</b></p> <pre>Router(config)# end</pre>	<p>Exits to privileged EXEC mode.</p>

### Examples—CSC-PE

The following example shows the configuration of /32 static routes from the CSC-PE router to the CSC-CE router’s loopback address:

```
configure terminal
ip route vrf vpn1 10.10.10.10 255.255.255 e1/0 168.192.0.2
ip route vrf vpn1 10.10.10.10 255.255.255 e0/0 168.192.2.2
```

#### Configuring /32 Static Routes to the EBGp Neighbor Loopback for the CSC-CE Router

Perform the following task to configure /32 static routes to the EBGp neighbor loopback for the CSC-CE router.

## SUMMARY STEPS—CSC-CE

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag tag**]
4. **end**

## DETAILED STEPS—CSC-CE

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><b>ip route</b> <i>prefix mask</i> {<i>ip-address</i>   <i>interface-type interface-number</i> [<i>ip-address</i>]} [<i>distance</i>] [<i>name</i>] [<b>permanent</b>] [<b>tag tag</b>]</p> <p><b>Example:</b> Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet1/0 168.192.0.1</p> <p>Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet0/0 168.192.2.1</p>	<p>Establishes static routes.</p> <ul style="list-style-type: none"> <li>• The <i>prefix</i> argument is the IP route prefix for the destination.</li> <li>• The <i>mask</i> argument is the prefix mask for the destination.</li> <li>• The <i>ip-address</i> argument is the IP address of the next hop that you can use to reach the destination network.</li> <li>• The <i>interface-type</i> and <i>interface-number</i> arguments are the network interface type and interface number.</li> <li>• The <i>distance</i> argument is an administrative distance.</li> <li>• The <i>name</i> argument applies a name to the specified route.</li> <li>• The <b>permanent</b> keyword specifies that the route is not to be removed, even if the interface shuts down.</li> <li>• The <b>tag tag</b> keyword-argument pair names a tag value that can be used as a “match” value for controlling redistribution via route maps.</li> </ul>
Step 4	<p><b>end</b></p> <p><b>Example:</b> Router(config)# end</p>	<p>Exits to privileged EXEC mode.</p>

## Examples—CSC-CE

The following example shows the configuration of /32 static routes from the CSC-CE router to the CSC-PE router's loopback address:

```
configure terminal
ip route 10.20.20.20 255.255.255 e1/0 168.192.0.1
ip route 10.20.20.20 255.255.255 e0/0 168.192.2.1
```

## Configuring Forwarding on Connecting Loopback Interfaces

Perform this task to configure forwarding on the connecting loopback interfaces.

This task is required for sessions between loopbacks. In the “[Configuring /32 Static Routes to the EBGp Neighbor Loopback](#)” task, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

### Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback

Perform this task to configure forwarding on CSC-PE interfaces that connect to the CSC-CE loopback.

## SUMMARY STEPS—CSC-PE

1. **enable**
2. **configure terminal**
3. **interface** *interface-type slot/port*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **mpls bgp forwarding**
7. **exit**
8. Repeat Steps 3 and 6 for another connecting interface (Ethernet 0/0)
9. **end**

## DETAILED STEPS—CSC-PE

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

	Command or Action	Purpose
Step 3	<pre>interface interface-type slot/port</pre> <p><b>Example:</b> Router(config)# interface ethernet 1/0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>interface-type</i> argument is the type of interface to be configured.</li> <li>The <i>slot</i> argument is the slot number. Refer to the appropriate hardware manual for slot and port information.</li> <li>The <i>/port</i> argument is the port number. Refer to the appropriate hardware manual for slot and port information.</li> </ul>
Step 4	<pre>ip vrf forwarding vrf-name</pre> <p><b>Example:</b> Router(config-if)# ip vrf forwarding vpn1</p>	<p>Associates a VRF with an interface or subinterface.</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
Step 5	<pre>ip address ip-address mask [secondary]</pre> <p><b>Example:</b> Router(config-if)# ip address 168.192.0.1 255.255.255.255</p>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address.</li> <li>The <i>mask</i> argument is the mask for the associated IP subnet.</li> <li>The <b>secondary</b> keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.</li> </ul>
Step 6	<pre>mpls bgp forwarding</pre> <p><b>Example:</b> Router(config-if)# mpls bgp forwarding</p>	<p>Configures BGP to enable MPLS forwarding on connecting interfaces.</p>
Step 7	<pre>exit</pre> <p><b>Example:</b> Router(config-if)# exit</p>	<p>Exits to global configuration mode.</p>
Step 8	<p>Repeat Steps 3 and 6 for another connecting interface (Ethernet 0/0).</p>	<p>—</p>
Step 9	<pre>end</pre> <p><b>Example:</b> Router(config)# end</p>	<p>Exits to privileged EXEC mode.</p>

### Examples—CSC-PE

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the CSC-PE router with the CSC-CE router:

```
configure terminal
interface ethernet 1/0
 ip vrf forwarding vpn1
 ip address 168.192.0.1 255.255.255.255
 mpls bgp forwarding
```

```

exit
interface ethernet 0/0
 ip vrf forwarding vpn1
 ip address 168.192.2.1 255.255.255.255
 mpls bgp forwarding
exit

```

### Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback

Perform this task to configure forwarding on CSC-CE interfaces that connect to the CSC-PE loopback.

#### SUMMARY STEPS—CSC-CE

1. **enable**
2. **configure terminal**
3. **interface** *interface-type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0)
7. **end**

#### DETAILED STEPS—CSC-CE

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>interface</b> <i>interface-type slot/port</i>  <b>Example:</b> Router(config)# interface ethernet 1/0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> <li>• The <i>interface-type</i> argument is the type of interface to be configured.</li> <li>• The <i>slot</i> argument is the slot number. Refer to the appropriate hardware manual for slot and port information.</li> <li>• The <i>/port</i> argument is the port number. Refer to the appropriate hardware manual for slot and port information.</li> </ul>
Step 4	<b>mpls bgp forwarding</b>  <b>Example:</b> Router(config-if)# mpls bgp forwarding	Configures BGP to enable MPLS forwarding on connecting interfaces.

	Command or Action	Purpose
Step 5	<code>exit</code>  <b>Example:</b> <code>Router(config-if)# exit</code>	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	—
Step 7	<code>end</code>  <b>Example:</b> <code>Router(config)# end</code>	Exits to privileged EXEC mode.

### Examples—CSC-CE

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the CSC-CE router with the CSC-PE router:

```
configure terminal
interface ethernet 1/0
  ip address 168.192.0.2 255.255.255.255
  mpls bgp forwarding
  exit
interface ethernet 0/0
  ip address 168.192.2.0 255.255.255.255
  mpls bgp forwarding
  exit
```

### Configuring an EBGP Session Between the Loopbacks

Perform the following tasks to configure an EBGP session between the loopbacks.

#### Configuring an EBGP Session Between the CSC-PE Router and the CSC-CE Loopback

Perform this task to configure an EBGP session between the CSC-PE router and the CSC-CE loopback.

### SUMMARY STEPS—CSC-PE

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
9. **ip vrf forwarding** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

11. **neighbor** *ip-address* **send-label**

12. **end**

## DETAILED STEPS—CSC-PE

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>router bgp</b> <i>as-number</i>  <b>Example:</b> Router(config)# router bgp 200	Configures the BGP routing process. <ul style="list-style-type: none"><li>The <i>as-number</i> indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.</li></ul>
Step 4	<b>bgp log-neighbor-changes</b>  <b>Example:</b> Router(config-router)# bgp log-neighbor-changes	Enables logging of BGP neighbor resets.
Step 5	<b>neighbor</b> { <i>ip-address</i>   <i>peer-group-name</i> } <b>remote-as</b> <i>as-number</i>  <b>Example:</b> Router(config-router)# neighbor 10.10.10.10 remote-as 100	Adds an entry to the BGP or multiprotocol BGP neighbor table. <ul style="list-style-type: none"><li>The <i>ip-address</i> argument is the IP address of the neighbor.</li><li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li><li>The <i>as-number</i> argument is the autonomous system to which the neighbor belongs.</li></ul>
Step 6	<b>neighbor</b> { <i>ip-address</i>   <i>peer-group-name</i> } <b>disable-connected-check</b>  <b>Example:</b> Router(config-router)# neighbor 10.10.10.10 disable-connected-check	Allows peering between loopbacks. <ul style="list-style-type: none"><li>The <i>ip-address</i> argument is the IP address of the neighbor.</li><li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li></ul>

Command or Action	Purpose
<p><b>Step 7</b></p> <pre>neighbor {ip-address   ipv6-address   peer-group-name} update-source interface-type interface-number</pre> <p><b>Example:</b> Router(config-router)# neighbor 10.10.10.10 update-source Loopback0</p>	<p>Allows BGP sessions in Cisco IOS releases to use any operational interface for TCP connections.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IPv4 address of the BGP-speaking neighbor.</li> <li>The <i>ipv6-address</i> argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.</li> <li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li> <li>The <i>interface-type</i> argument is the interface type.</li> <li>The <i>interface-number</i> argument is the interface number.</li> </ul>
<p><b>Step 8</b></p> <pre>address-family ipv4 [unicast] vrf vrf-name</pre> <p><b>Example:</b> Router(config-router)# address-family ipv4 vrf vpn1</p>	<p>Enters the address family submode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing.</p> <ul style="list-style-type: none"> <li>The <b>ipv4</b> keyword configures sessions that carry standard IPv4 address prefixes.</li> <li>The <b>unicast</b> keyword specifies unicast prefixes.</li> <li>The <b>vrf vrf-name</b> keyword-argument pair specifies the name of a VRF to associate with submode commands.</li> </ul>
<p><b>Step 9</b></p> <pre>ip vrf forwarding vrf-name</pre> <p><b>Example:</b> Router(config-router-af)# ip vrf forwarding vpn1</p>	<p>Associates a VRF with an interface or subinterface.</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
<p><b>Step 10</b></p> <pre>neighbor {ip-address   peer-group-name   ipv6-address} activate</pre> <p><b>Example:</b> Router(config-router-af)# neighbor 10.10.10.10 activate</p>	<p>Enables the exchange of information with a BGP neighbor.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighboring router.</li> <li>The <i>peer-group-name</i> argument is the name of the BGP peer group.</li> <li>The <i>ipv6-address</i> argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.</li> </ul>



	Command or Action	Purpose
Step 11	<pre>neighbor ip-address send-label</pre> <p><b>Example:</b> Router(config-router-af)# neighbor 10.10.10.10 send-label</p>	<p>Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighboring router.</li> </ul>
Step 12	<pre>end</pre> <p><b>Example:</b> Router(config)# end</p>	<p>Exits to privileged EXEC mode.</p>

### Examples—CSC-PE

The example below shows the required configuration for IPv4 sessions on a CSC-PE router:

```
router bgp 200
  bgp log-neighbor-changes
  neighbor 10.10.10.10 remote-as 100
  neighbor 10.10.10.10 disable-connected-check
  neighbor 10.10.10.10 update-source Loopback0
  !
  address-family ipv4 vrf vpn1
  ip vrf forwarding vpn1
  neighbor 10.10.10.10 activate
  neighbor 10.10.10.10 send-label
  !
```

#### Configuring an EBGP Session Between the CSC-CE Router and the CSC-PE Loopback

Perform this task to configure an EBGP session between the CSC-CE router and the CSC-PE loopback.

### SUMMARY STEPS—CSC-CE

- enable**
- configure terminal**
- router bgp** *as-number*
- bgp log-neighbor-changes**
- neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
- neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
- address-family ipv4** [**unicast**] [**vrf** *vrf-name*]
- neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
- neighbor** *ip-address* **send-label**
- end**

## DETAILED STEPS—CSC-CE

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>router bgp as-number</b>  <b>Example:</b> Router(config)# router bgp 200	Configures the BGP routing process. <ul style="list-style-type: none"> <li>The <i>as-number</i> indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.</li> </ul>
Step 4	<b>bgp log-neighbor-changes</b>  <b>Example:</b> Router(config-router)# bgp log-neighbor-changes	Enables logging of BGP neighbor resets.
Step 5	<b>neighbor {ip-address   peer-group-name}</b> <b>remote-as as-number</b>  <b>Example:</b> Router(config-router)# neighbor 10.20.20.20 remote-as 100	Adds an entry to the BGP or multiprotocol BGP neighbor table. <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li> <li>The <i>as-number</i> argument is the autonomous system to which the neighbor belongs.</li> </ul>
Step 6	<b>neighbor {ip-address   peer-group-name}</b> <b>disable-connected-check</b>  <b>Example:</b> Router(config-router)# neighbor 10.20.20.20 disable-connected-check	Allows peering between loopbacks. <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li> </ul>

	Command or Action	Purpose
Step 7	<pre>neighbor {ip-address   ipv6-address   peer-group-name} update-source interface-type interface-number</pre> <p><b>Example:</b> Router(config-router)# neighbor 10.20.20.20 update-source Loopback0</p>	<p>Allows BGP sessions in Cisco IOS releases to use any operational interface for TCP connections.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IPv4 address of the BGP-speaking neighbor.</li> <li>The <i>ipv6-address</i> argument is the IPv6 address of the BGP-speaking neighbor.</li> </ul> <p>This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.</p> <ul style="list-style-type: none"> <li>The <i>peer-group-name</i> argument is the name of a BGP peer group.</li> <li>The <i>interface-type</i> argument is the interface type.</li> <li>The <i>interface-number</i> argument is the interface number.</li> </ul>
Step 8	<pre>address-family ipv4 [unicast] [vrf vrf-name]</pre> <p><b>Example:</b> Router(config-router)# address-family ipv4</p>	<p>Enters the address family submode for configuring routing protocols such as BGP, RIP, and static routing.</p> <ul style="list-style-type: none"> <li>The <b>ipv4</b> keyword configures sessions that carry standard IPv4 address prefixes.</li> <li>The <b>unicast</b> keyword specifies unicast prefixes.</li> <li>The <b>vrf vrf-name</b> keyword-argument pair specifies the name of a VRF to associate with submode commands.</li> </ul>
Step 9	<pre>neighbor {ip-address   peer-group-name / ipv6-address} activate</pre> <p><b>Example:</b> Router(config-router-af)# neighbor 10.20.20.20 activate</p>	<p>Enables the exchange of information with a BGP neighbor.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighboring router.</li> <li>The <i>peer-group-name</i> argument is the name of the BGP peer group.</li> <li>The <i>ipv6-address</i> argument is the IPv6 address of the BGP-speaking neighbor.</li> </ul> <p>This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.</p>
Step 10	<pre>neighbor ip-address send-label</pre> <p><b>Example:</b> Router(config-router-af)# neighbor 10.20.20.20 send-label</p>	<p>Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address of the neighboring router.</li> </ul>
Step 11	<pre>end</pre> <p><b>Example:</b> Router(config)# end</p>	<p>Exits to privileged EXEC mode.</p>

## Examples—CSC-CE

The example below shows the required configuration for IPv4 sessions on a CSC-PE router:

```
router bgp 200
  bgp log-neighbor-changes
  neighbor 10.20.20.20 remote-as 100
  neighbor 10.20.20.20 disable-connected-check
  neighbor 10.20.20.20 update-source Loopback0
  !
  address-family ipv4
  neighbor 10.20.20.20 activate
  neighbor 10.20.20.20 send-label
  !
```

## Verifying That Load Balancing Occurs Between Loopbacks

To verify that load balancing occurs between loopbacks, ensure that the MPLS Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

### SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
3. **disable**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>show mpls forwarding-table</b> [ <b>vrf</b> <i>vrf-name</i> ] [{ <i>network</i> { <i>mask</i>   <i>length</i> }   <b>labels</b> <i>label</i> [- <i>label</i> ]   <b>interface</b> <i>interface</i>   <b>next-hop</b> <i>address</i>   <b>lsp-tunnel</b> [ <i>tunnel-id</i> ]}] [ <b>detail</b> ]  <b>Example:</b> Router# show mpls forwarding-table	Displays the contents of the MPLS LFIB.
Step 3	<b>disable</b>  <b>Example:</b> Router# disable	Exits to user EXEC mode.

## Configuring Route Maps on the CSC-PE Routers

Perform the following tasks to enable routers to send MPLS labels with the routes specified in the route maps:

- [Configuring a Route Map for Incoming Routes, page 45](#) (optional)
- [Configuring a Route Map for Outgoing Routes, page 48](#) (optional)
- [Applying the Route Maps to the CSC-PE and CSC-CE Routers, page 50](#) (optional)

To configure route maps on routers, specifically carrier edge routers, you need to understand how to use route maps to filter routes.

### Using Route Maps to Filter Routes

When routers are configured to distribute routes with MPLS labels, all the routes are encoded with the multiprotocol extensions and contain MPLS labels. You can use a route map to control the distribution of MPLS labels between routers.

Route maps enable you to specify which routes are distributed with MPLS labels. Route maps also enable you to specify which routes with MPLS labels a router receives and adds to its BGP table.

Route maps work with access control lists (ACLs). You enter the routes into an ACL and then specify the ACL when you configure the route map. You can configure a router to accept only routes that are specified in the route map. The router checks the routes listed in the BGP update message against the list of routes in the specified ACL. If a route in the BGP update message matches a route in the ACL, the route is accepted and added to the BGP table.

### Prerequisites

Before you configure and apply route maps for the CSC-PE routers, you need to create an ACL and specify the routes that the router should distribute with MPLS labels.

### Configuring a Route Map for Incoming Routes

This configuration is optional.

Perform this task to create a route map to filter arriving routes. You create an access list and specify the routes that the router should accept and add to the BGP table.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **route-map** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** { *access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*] *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]
6. **match mpls-label**
7. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>router bgp as-number</pre> <p><b>Example:</b> Router(config)# router bgp 100 </p>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>
Step 4	<pre>route-map map-name [permit   deny] sequence-number</pre> <p><b>Example:</b> Router(config-router)# route-map csc-mpls-routes-in permit </p>	<p>Enters route map configuration mode and creates a route map with the name you specify.</p> <ul style="list-style-type: none"> <li>The <i>map-name</i> argument identifies the name of the route map.</li> <li>The <b>permit</b> keyword allows the actions to happen if all conditions are met.</li> <li>A <b>deny</b> keyword prevents any actions from happening if all conditions are met.</li> <li>The <i>sequence-number</i> argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on.</li> </ul>

Command or Action	Purpose
<p><b>Step 5</b></p> <pre>match ip address {access-list-number [access-list-number...   access-list-name...]   access-list-name [access-list-number...   access-list-name]   prefix-list prefix-list-name [prefix-list-name...]}</pre> <p><b>Example:</b> Router(config-route-map)# match ip address acl-in</p>	<p>Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets.</p> <ul style="list-style-type: none"> <li>• The <i>access-list-number...</i> argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.</li> <li>• The <i>access-list-name...</i> argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.</li> <li>• The <b>prefix-list</b> keyword distributes routes based on a prefix list.</li> <li>• The <i>prefix-list-name...</i> argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.</li> </ul>
<p><b>Step 6</b></p> <pre>match mpls-label</pre> <p><b>Example:</b> Router(config-route-map)# match mpls-label</p>	<p>Redistributes routes that include MPLS labels if the routes meet the conditions specified in the route map.</p>
<p><b>Step 7</b></p> <pre>exit</pre> <p><b>Example:</b> Router(config-router-map)# exit</p>	<p>Exits route map configuration mode and returns to global configuration mode.</p>

## Configuring a Route Map for Outgoing Routes

This configuration is optional.

Perform this task to create a route map to filter departing routes. You create an access list and specify the routes that the router should distribute with MPLS labels.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **route-map** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** { *access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **set mpls-label**
7. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>router bgp</b> <i>as-number</i>  <b>Example:</b> Router(config)# router bgp 100	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> <li>• The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.</li> </ul> <p>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.</p>



	Command or Action	Purpose
Step 4	<pre>route-map map-name [permit   deny] sequence-number</pre> <p><b>Example:</b> Router(config-router)# route-map csc-mpls-routes-out permit</p>	<p>Enters route map configuration mode and creates a route map with the name you specify.</p> <ul style="list-style-type: none"> <li>The <i>map-name</i> argument identifies the name of the route map.</li> <li>The <b>permit</b> keyword allows the actions to happen if all conditions are met.</li> <li>A <b>deny</b> keyword prevents any actions from happening if all conditions are met.</li> <li>The <i>sequence-number</i> argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on.</li> </ul>
Step 5	<pre>match ip address {access-list-number [access-list-number...   access-list-name...]   access-list-name [access-list-number...   access-list-name]   prefix-list prefix-list-name [prefix-list-name...]}</pre> <p><b>Example:</b> Router(config-route-map)# match ip address acl-out</p>	<p>Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets.</p> <ul style="list-style-type: none"> <li>The <i>access-list-number...</i> argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.</li> <li>The <i>access-list-name...</i> argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.</li> <li>The <b>prefix-list</b> keyword distributes routes based on a prefix list.</li> <li>The <i>prefix-list-name...</i> argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.</li> </ul>
Step 6	<pre>set mpls-label</pre> <p><b>Example:</b> Router(config-route-map)# set mpls-label</p>	<p>Enables a route to be distributed with an MPLS label if the route matches the conditions specified in the route map.</p>
Step 7	<pre>exit</pre> <p>Router(config-router-map)# exit</p>	<p>Exits route map configuration mode and returns to global configuration mode.</p>

## Applying the Route Maps to the CSC-PE and CSC-CE Routers

This configuration is optional.

Perform this task to enable the CSC-PE and the CSC-CE routers to use the route maps.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **route-map** *route-map-name* **in**
6. **neighbor** *ip-address* **route-map** *route-map-name* **out**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **end**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>router bgp</b> <i>as-number</i>  <b>Example:</b> Router(config)# router bgp 100	Configures a BGP routing process and enters router configuration mode. <ul style="list-style-type: none"> <li>• The <i>as-number</i> 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.</li> </ul>

	Command or Action	Purpose
Step 4	<pre>address-family ipv4 [multicast   unicast   vrf vrf-name]</pre> <p><b>Example:</b> Router(config-router)# address-family ipv4 vrf vpn1</p>	<p>Specifies the IPv4 address family type and enters address family configuration mode.</p> <ul style="list-style-type: none"> <li>The <b>multicast</b> keyword specifies IPv4 multicast address prefixes.</li> <li>The <b>unicast</b> keyword specifies IPv4 unicast address prefixes.</li> <li>The <b>vrf vrf-name</b> keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</li> </ul>
Step 5	<pre>neighbor ip-address route-map map-name in</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 route-map csc-mpls-routes-in in</p>	<p>Applies a route map to incoming routes.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the router to which the route map is to be applied.</li> <li>The <i>map-name</i> argument specifies the name of the route map.</li> <li>The <b>in</b> keyword applies the route map to incoming routes.</li> </ul>
Step 6	<pre>neighbor ip-address route-map map-name out</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 route-map csc-mpls-route-out out</p>	<p>Applies a route map to outgoing routes.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the router to which the route map is to be applied.</li> <li>The <i>map-name</i> argument specifies the name of the route map.</li> <li>The <b>out</b> keyword applies the route map to outgoing routes.</li> </ul>
Step 7	<pre>neighbor ip-address send-label</pre> <p><b>Example:</b> Router(config-router-af)# neighbor pp.0.0.1 send-label</p>	<p>Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighboring router.</li> </ul>
Step 8	<pre>exit-address-family</pre> <p><b>Example:</b> Router(config-router-af)# exit-address-family</p>	<p>Exits from the address family configuration mode.</p>
Step 9	<pre>end</pre> <p><b>Example:</b> Router(config-router)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

### Troubleshooting Tips

You can enter a **show route-map map-name** command to verify that the route map is applied to the CSC-CE and the CSC-PE routers.

**Note**

After you make any changes to a route map, you need to reset the BGP connection for the changes to take effect.

**Additional Information**

For a configuration example for this task, see the [“Configuring Route Maps on the CSC-PE Routers: Example”](#) section on page 84.

## Configuring and Verifying the Customer Carrier Network

Perform the following tasks to configure and verify the customer carrier network. This requires setting up connectivity and routing functions for the customer carrier core (P) routers and the customer carrier edge (PE) routers.

- [Verifying IP Connectivity in the Customer Carrier, page 52](#) (optional)
- [Configuring a Customer Carrier Core Router as a Route Reflector, page 53](#) (optional)

### Prerequisites

Before you configure a customer carrier network for the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature, you must configure the following on your customer carrier routers:

- An IGP routing protocol—BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see the “IP Routing Protocols” chapter in the [Cisco IOS IP Configuration Guide, Release 12.0](#).
- MPLS VPN functionality on the PE routers (for hierarchical VPNs only). For information, see the [MPLS Virtual Private Networks \(VPNs\)](#) or the [MPLS Virtual Private Network Enhancements](#).
- Label Distribution Protocol (LDP) on P and PE routers (for hierarchical VPNs only). For information, see the [MPLS Label Distribution Protocol \(LDP\)](#).

**Note**

You must configure the items in the preceding list before performing the tasks in this section.

### Verifying IP Connectivity in the Customer Carrier

Perform this task to verify IP connectivity in the customer carrier.

**SUMMARY STEPS**

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route**
5. **disable**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<code>ping [protocol] {host-name   system-address}</code>  <b>Example:</b> Router# <code>ping ip &lt;P-address&gt;</code>	Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. <ul style="list-style-type: none"> <li>Use the <b>ping</b> command to verify the connectivity from one customer carrier core router to another.</li> </ul>
Step 3	<code>trace [protocol] [destination]</code>  <b>Example:</b> Router# <code>trace ip destination-address</code>	Discovers the routes that packets will actually take when traveling to their destination. <ul style="list-style-type: none"> <li>Use the <b>trace</b> command to verify the path that a packet goes through before reaching the final destination. The <b>trace</b> command can help isolate a trouble spot if two routers cannot communicate.</li> </ul>
Step 4	<code>show ip route</code>  <b>Example:</b> Router# <code>show ip route</code>	Displays IP routing table entries. <ul style="list-style-type: none"> <li>Use the <b>show ip route</b> command to display the entire routing table, including host IP address, next hop, interface, and so forth.</li> </ul>
Step 5	<code>disable</code>  <b>Example:</b> Router# <code>disable</code>	Returns to user mode.

## Additional Information

For a configuration example for this task, see the [“Verifying IP Connectivity in the Customer Carrier: Example”](#) section on page 84.

## Configuring a Customer Carrier Core Router as a Route Reflector

Perform this task to configure a customer carrier core (P) router as a route reflector of multiprotocol BGP prefixes.

## SUMMARY STEPS

- `enable`
- `configure terminal`
- `router bgp as-number`
- `neighbor {ip-address | peer-group-name} remote-as as-number`
- `address-family vpnv4 [unicast]`
- `neighbor {ip-address | peer-group-name} activate`

7. **neighbor *ip-address* route-reflector-client**
8. **exit-address-family**
9. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>router bgp <i>as-number</i></pre> <p><b>Example:</b> Router(config)# router bgp 200 </p>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> argument indicates the number of an autonomous system that identifies the router to other BGP routers and labels 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.</li> </ul>
Step 4	<pre>neighbor {<i>ip-address</i>   <i>peer-group-name</i>} remote-as <i>as-number</i></pre> <p><b>Example:</b> Router(config-router)# neighbor 10.1.1.1 remote-as 100 </p>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>
Step 5	<pre>address-family vpnv4 [<b>unicast</b>]</pre> <p><b>Example:</b> Router(config-router)# address-family vpnv4 </p>	<p>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.</p> <ul style="list-style-type: none"> <li>The optional <b>unicast</b> keyword specifies VPNv4 unicast address prefixes.</li> </ul>
Step 6	<pre>neighbor {<i>ip-address</i>   <i>peer-group-name</i>} activate</pre> <p><b>Example:</b> Router(config-router-af)# neighbor 10.1.1.1 activate </p>	<p>Enables the exchange of information with a neighboring BGP router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>

	Command or Action	Purpose
Step 7	<pre>neighbor ip-address route-reflector-client</pre> <p><b>Example:</b>  Router(config-router-af)# neighbor 10.1.1.1 route-reflector-client</p>	Configures the router as a BGP route reflector and configures the specified neighbor as its client. <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the BGP neighbor being identified as a client.</li> </ul>
Step 8	<pre>exit-address-family</pre> <p><b>Example:</b>  Router(config-router-af)# exit-address-family</p>	Exits address family configuration mode.
Step 9	<pre>end</pre> <p><b>Example:</b>  Router(config-router)# end</p>	(Optional) Exits to privileged EXEC mode.

### Troubleshooting Tips

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. For neighbors to exchange other address prefix types, such as multicast and VPNv4, you must also activate neighbors using the **neighbor activate** command in address family configuration mode, as shown.

Route reflectors and clients (neighbors or internal BGP peer groups) that are defined in router configuration mode using the **neighbor route-reflector-client** command reflect unicast address prefixes to and from those clients by default. To cause them to reflect prefixes for other address families, such as multicast, define the reflectors and clients in address family configuration mode, using the **neighbor route-reflector-client** command, as shown.

### Additional Information

For a configuration example for this task, see the [“Configuring a Customer Carrier Core Router as a Route Reflector: Example”](#) section on page 85.

## Configuring and Verifying the Customer Site for Hierarchical VPNs

Perform the following tasks to configure and verify the customer site for hierarchical VPNs:

- [Configuring Provider Edge Routers for Hierarchical VPNs, page 56](#) (required)
- [Verifying Labels in Each Provider Edge Router for Hierarchical VPNs, page 59](#) (optional)
- [Configuring CE Routers for Hierarchical VPNs, page 60](#) (required)
- [Verifying IP Connectivity in the Customer Site, page 62](#) (optional)



Note

This section applies to hierarchical VPNs only.

## Configuring Provider Edge Routers for Hierarchical VPNs

Perform the following tasks to configure PE routers for hierarchical VPNs:

- [Defining VPNs on PE Routers, page 56](#) (required)
- [Configuring BGP Routing Sessions on the Provider Edge Routers, page 57](#) (required)

### Defining VPNs on PE Routers

Perform this task to define VPNs on PE routers.

#### 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. **ip vrf forwarding** *vrf-name*
8. **exit**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>ip vrf</b> <i>vrf-name</i>  <b>Example:</b> Router(config)# ip vrf vpn2	Creates a VRF routing table and a CEF forwarding table and enters VRF configuration mode.  • The <i>vrf-name</i> argument is a name you assign to a VRF.
Step 4	<b>rd</b> <i>route-distinguisher</i>  <b>Example:</b> Router(config-vrf)# rd 200:1	Creates routing and forwarding tables for a VRF.  • The <i>route-distinguisher</i> argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.



	Command or Action	Purpose
Step 5	<pre>route-target {import   export   both} route-target-ext-community</pre> <p><b>Example:</b> Router(config-vrf)# route-target export 200:1</p>	<p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> <li>The <b>import</b> keyword imports routing information from the target VPN extended community.</li> <li>The <b>export</b> keyword exports routing information to the target VPN extended community.</li> <li>The <b>both</b> keyword imports routing information from and export routing information to the target VPN extended community.</li> <li>The <i>route-target-ext-community</i> 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.</li> </ul>
Step 6	<pre>import map route-map</pre> <p><b>Example:</b> Router(config-vrf)# import map &lt;route-map&gt;</p>	<p>Configures an import route map for a VRF.</p> <ul style="list-style-type: none"> <li>The <i>route-map</i> argument specifies the route map to be used as an import route map for the VRF.</li> </ul>
Step 7	<pre>ip vrf forwarding vrf-name</pre> <p><b>Example:</b> Router(config-vrf)# ip vrf forwarding vpn2</p>	<p>Associates a VPN VRF instance with an interface or subinterface.</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>
Step 8	<pre>exit</pre> <p><b>Example:</b> Router(config-vrf)# exit</p>	<p>Exits to global configuration mode.</p>

### Additional Information

For a configuration example for this task, see the [“Configuring Provider Edge Routers for Hierarchical VPNs: Examples”](#) section on page 86.

### Configuring BGP Routing Sessions on the Provider Edge Routers

Perform this task to configure BGP routing sessions on the PE routers for PE-to-CE router communication.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal </p>	<p>Enters global configuration mode.</p>
Step 3	<pre>router bgp as-number</pre> <p><b>Example:</b> Router(config)# router bgp 200 </p>	<p>Configures the router to run a BGP process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>
Step 4	<pre>address-family ipv4 [multicast   unicast   vrf vrf-name]</pre> <p><b>Example:</b> Router(config-router)# address-family ipv4 multicast </p>	<p>Specifies the IPv4 address family type and enters address family configuration mode.</p> <ul style="list-style-type: none"> <li>The <b>multicast</b> keyword specifies IPv4 multicast address prefixes.</li> <li>The <b>unicast</b> keyword specifies IPv4 unicast address prefixes.</li> <li>The <b>vrf vrf-name</b> keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.</li> </ul>
Step 5	<pre>neighbor {ip-address   peer-group-name} remote-as as-number</pre> <p><b>Example:</b> Router(config-router-af)# neighbor &lt;remote-PE-IP-address&gt; remote-as 300 </p>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>

	Command or Action	Purpose
Step 6	<pre>neighbor {ip-address   peer-group-name} activate</pre> <p><b>Example:</b>  Router(config-router-af)# neighbor  &lt;CE-IP-address&gt; activate</p>	<p>Enables the exchange of information with a neighboring router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> </ul>
Step 7	<pre>end</pre> <p><b>Example:</b>  Router(config-router-af)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

### Additional Information

For a configuration example for this task, see the “[Configuring Provider Edge Routers for Hierarchical VPNs: Examples](#)” section on page 86.

## Verifying Labels in Each Provider Edge Router for Hierarchical VPNs

Perform this task to verify labels in each PE router for hierarchical VPNs.

### SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**]
4. **show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**]
5. **show ip cef vrf** *vrf-name* [*ip-prefix*]
6. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b>  Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>show ip route vrf vrf-name [prefix]</pre> <p><b>Example:</b>  Router# show ip route vrf vpn2 &lt;CE-prefix&gt;</p>	<p>(Optional) Displays the IP routing table associated with a VRF.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip route vrf</b> command to check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.</li> </ul>

	Command or Action	Purpose
Step 3	<pre>show mpls forwarding-table [vrf vrf-name] [<i>prefix</i>] [<i>detail</i>]</pre> <p><b>Example:</b>  Router# show mpls forwarding-table vrf vpn2  &lt;CE-prefix&gt;</p> <pre>Router# show mpls forwarding-table vrf vpn2 &lt;CE-prefix&gt; detail</pre>	(Optional) Displays the contents of the LFIB. <ul style="list-style-type: none"> <li>Use the <b>show mpls forwarding-table</b> command to check that the prefixes for the local and remote CE routers are in the MPLS forwarding table, and that the specified prefix is untagged.</li> </ul>
Step 4	<pre>show ip cef [<i>network</i> [<i>mask</i> [<i>longer-prefix</i>]]] [<i>detail</i>]</pre> <p><b>Example:</b>  Router# show ip cef &lt;PE-prefix&gt;</p> <pre>Router# show ip cef &lt;PE-prefix&gt; detail</pre>	(Optional) Displays specific entries in the FIB based on IP address information. <ul style="list-style-type: none"> <li>Use the <b>show ip cef</b> command to check that the prefixes of the local and remote PE routers are in the CEF table.</li> </ul>
Step 5	<pre>show ip cef vrf vrf-name [<i>ip-prefix</i>]</pre> <p><b>Example:</b>  Router# show ip cef vrf vpn2 &lt;CE-prefix&gt;</p>	(Optional) Displays the CEF forwarding table associated with a VRF. <ul style="list-style-type: none"> <li>Use the <b>show ip cef vrf</b> command to check that the prefix of the remote CE router is in the CEF table.</li> </ul>
Step 6	<pre>exit</pre> <p><b>Example:</b>  Router# exit</p>	(Optional) Exits to user EXEC mode.

### Additional Information

For a configuration example for this task, see the [“Verifying Labels in Each Provider Edge Router for Hierarchical VPNs: Examples”](#) section on page 88.

## Configuring CE Routers for Hierarchical VPNs

Perform this task to configure CE routers for hierarchical VPNs. This configuration is the same as that for an MPLS VPN that is not in a hierarchical topology.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef [distributed]**
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **router bgp** *as-number*
7. **redistribute** *protocol*
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<pre>ip cef [distributed]</pre> <p><b>Example:</b> Router(config)# ip cef distributed</p>	<p>Enables CEF on the route processor card.</p> <ul style="list-style-type: none"> <li>The <b>distributed</b> keyword enables distributed CEF (dCEF) operation. CEF information is distributed to the line cards. Line cards perform express forwarding.</li> </ul>
Step 4	<pre>interface type number</pre> <p><b>Example:</b> Router(config)# interface loopback 0</p>	<p>Configures an interface type and enters interface configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>type</i> argument specifies the type of interface to be configured. <ul style="list-style-type: none"> <li>A loopback interface indicates a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms.</li> </ul> </li> <li>The <i>number</i> argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.</li> </ul>
Step 5	<pre>ip address ip-address mask [secondary]</pre> <p><b>Example:</b> Router(config-if)# ip address aa.aa.aa.aa 255.255.2355.255</p>	<p>Sets a primary or secondary IP address for an interface.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument is the IP address.</li> <li>The <i>mask</i> argument is the mask for the associated IP subnet.</li> <li>The <b>secondary</b> keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.</li> </ul>
Step 6	<pre>router bgp as-number</pre> <p><b>Example:</b> Router(config-af)# router bgp 100</p>	<p>Configures a BGP routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>as-number</i> 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.</li> </ul>

	Command or Action	Purpose
Step 7	<pre>redistribute protocol</pre> <p><b>Example:</b> Router(config-router)# redistribute connected</p>	<p>Redistributes routes from one routing domain into another routing domain.</p> <ul style="list-style-type: none"> <li>The <i>protocol</i> argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: <b>bgp</b>, <b>connected</b>, <b>egp</b>, <b>igrp</b>, <b>isis</b>, <b>mobile</b>, <b>ospf</b>, <b>static [ip]</b>, or <b>rip</b>.</li> </ul> <p>The <b>connected</b> keyword refers to routes that are established automatically when IP is enabled on an interface. For routing protocols such as Open Shortest Path First (OSPF) and IS-IS, these routes are redistributed as external to the autonomous system.</p>
Step 8	<pre>neighbor {ip-address   peer-group-name} remote-as as-number</pre> <p><b>Example:</b> Router(config-router)# neighbor &lt;CSC-Core-ip-address&gt; remote-as 100</p>	<p>Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router.</p> <ul style="list-style-type: none"> <li>The <i>ip-address</i> argument specifies the IP address of the neighbor.</li> <li>The <i>peer-group-name</i> argument specifies the name of a BGP peer group.</li> <li>The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs.</li> </ul>
Step 9	<pre>end</pre> <p><b>Example:</b> Router(config-vrf)# end</p>	<p>(Optional) Exits to privileged EXEC mode.</p>

### Additional Information

For a configuration example for this task, see the [“Configuring Customer Edge Routers for Hierarchical VPNs: Examples”](#) section on page 92.

## Verifying IP Connectivity in the Customer Site

Perform this task to verify IP connectivity in the customer site.

### SUMMARY STEPS

1. **enable**
2. **show ip route** *[ip-address [mask] [longer-prefixes]]* | *[protocol [process-id]]* | **[list {access-list-number | access-list-name}]**
3. **ping** *[protocol] {host-name | system-address}*
4. **trace** *[protocol] [destination]*
5. **disable**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<pre>show ip route [ip-address [mask] [longer-prefixes]]   [protocol [process-id]]   [list {access-list-number   access-list-name}]</pre> <p><b>Example:</b> Router# show ip route &lt;remote-CE-address&gt;</p>	<p>(Optional) Displays the current state of the routing table.</p> <ul style="list-style-type: none"> <li>Use the <b>show ip route ip-address</b> command to check that the loopback addresses of the remote CE routers learned through the PE router are in the routing table of the local CE routers.</li> </ul>
Step 3	<pre>ping [protocol] {host-name   system-address}</pre> <p><b>Example:</b> Router# ping &lt;ip address&gt;</p>	<p>Diagnoses basic network connectivity on Apollo, AppleTalk, Connectionless Network Service (CLNS), DECnet, IP, Novell IPX, VINES, or XNS networks.</p> <ul style="list-style-type: none"> <li>Use the <b>ping</b> command to check connectivity between customer site routers.</li> </ul>
Step 4	<pre>trace [protocol] [destination]</pre> <p><b>Example:</b> Router# trace ip &lt;destination-ip-address&gt;</p>	<p>Discovers the routes that packets will actually take when traveling to their destination.</p> <ul style="list-style-type: none"> <li>Use the <b>trace</b> command to follow the path of the packets in the customer site.</li> <li>To use nondefault parameters and invoke an extended <b>trace</b> test, enter the <b>trace</b> command without a destination argument. You will be stepped through a dialog to select the desired parameters.</li> </ul>
Step 5	<pre>disable</pre> <p><b>Example:</b> Router# disable</p>	<p>(Optional) Exits to use EXEC mode.</p>

## Additional Information

For a configuration example for this task, see the [“Verifying IP Connectivity in the Customer Site: Examples”](#) section on page 93.

# Configuration Examples for MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution

Configuration examples for the MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution feature include the following:

- [Configuring and Verifying the Backbone Carrier Core: Examples, page 65](#)
- [Configuring and Verifying the Links Between CSC-PE and CSC-CE Routers: Examples, page 68](#)
- [Configuring and Verifying the Customer Carrier Network: Examples, page 84](#)
- [Configuring and Verifying the Customer Site for Hierarchical VPNs: Examples, page 86](#)

Figure 5 shows a sample CSC topology for exchanging IPv4 routes and MPLS labels. Use this figure as a reference for configuring and verifying carrier supporting carrier routers to exchange IPv4 routes and MPLS labels.

Figure 5 Sample CSC Topology for Exchanging IPv4 Routes and MPLS Labels

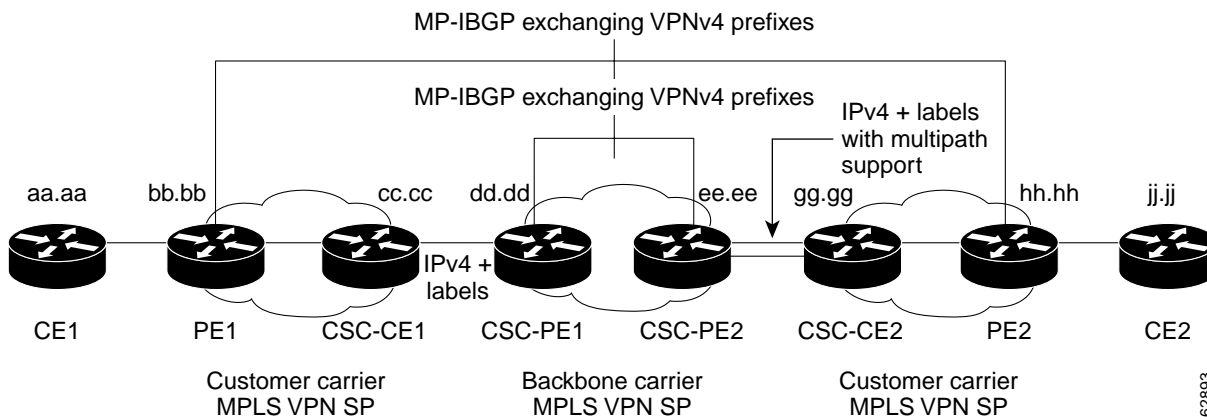


Table 2 describes the sample configuration shown in Figure 5.

Table 2 Description of Sample Configuration Shown in Figure 5

Routers	Description
CE1 and CE2	Belong to an end customer. CE1 and CE2 routers exchange routes learned from PE routers. The end customer is purchasing VPN services from a customer carrier.
PE1 and PE2	Part of a customer carrier network that is configured to provide MPLS VPN services. PE1 and PE2 are peering with a VPNv4 IBGP session to form an MPLS VPN network.



**Table 2** Description of Sample Configuration Shown in [Figure 5](#) (continued)

Routers	Description
CSC-CE1 and CSC-CE2	Part of a customer carrier network. CSC-CE1 and CSC-CE2 routers exchange IPv4 BGP updates with MPLS labels and redistribute PE loopback addressees to and from the IGP (OSPF in this example).  The customer carrier is purchasing carrier supporting carrier VPN services from a backbone carrier.
CSC-PE1 and CSC-PE2	Part of the backbone carrier's network configured to provide carrier supporting carrier VPN services. CSC-PE1 and CSC-PE2 are peering with a VPNv4 IP BGP session to form the MPLS VPN network. In the VRF, CSC-PE1 and CSC-PE2 are peering with the CSC-CE routers, which are configured for carrying MPLS labels with the routes, with an IPv4 EBGP session.

## Configuring and Verifying the Backbone Carrier Core: Examples

Configuration and verification examples for the backbone carrier core included in this section are as follows:

- [Verifying IP Connectivity and LDP Configuration in the CSC-Core: Example, page 65](#)
- [Configuring VRFs for CSC-PE Routers: Example, page 67](#)
- [Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier: Example, page 67](#)

### Verifying IP Connectivity and LDP Configuration in the CSC-Core: Example

Check that CSC-PE2 is reachable from CSC-PE1 by entering the following command on CSC-CE1:

```
Router# ping ee.ee.ee.ee

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ee.ee.ee.ee, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Verify the path from CSC-PE1 to CSC-PE2 by entering the following command on CSC-CE1:

```
Router# trace ee.ee.ee.ee

Type escape sequence to abort.
Tracing the route to ee.ee.ee.ee

  1 ee.ee.ee.ee 0 msec 0 msec *
```

Check that CSC-PE router prefixes are in the MPLS forwarding table:

```
Router# show mpls ip forwarding-table

Local   Outgoing   Prefix or   Bytes tag   Outgoing   Next Hop
tag     tag or VC  Tunnel Id   switched   interface
16      2/nn       dd.dd.dd.dd/32    0          AT2/1/0.1  point2point
17      16         bb.bb.bb.bb/32 [V] 30204      Et1/0      pp.0.0.1
21      Pop tag    cc.cc.cc.cc/32 [V] 0          Et1/0      pp.0.0.1
22      Pop tag    nn.0.0.0/8 [V]   570        Et1/0      pp.0.0.1
23      Aggregate  pp.0.0.0/8 [V]   0          Et1/0      pp.0.0.1
2       2/nn       gg.gg.gg.gg/32 [V] 0          AT3/0.1    point2point
```

```

8          2/nn          hh.hh.hh.hh/32 [V]  15452    AT3/0.1    point2point
29         2/nn          qq.0.0.0/8 [V]     0        AT3/0.1    point2point
30         2/nn          ss.0.0.0/8 [V]     0        AT3/0.1    point2point

```

Check the status of LDP discovery processes in the core:

```
Router# show mpls ldp discovery
```

```

Local LDP Identifier:
  ee.aa.bb.cc:0
Discovery Sources:
Interfaces:
  ATM2/1/0.1 (ldp): xmit/rcv
  TDP Id: dd.dd.dd.dd:1

```

Check the status of LDP sessions in the core:

```
Router# show mpls ldp neighbor
```

```

Peer LDP Ident: dd.dd.dd.dd:1; Local LDP Ident ee.aa.bb.cc:1
TCP connection: dd.dd.dd.dd.646 - ee.aa.bb.cc.11007
State: Oper; Msgs sent/rcvd: 20/21; Downstream on demand
Up time: 00:14:56
LDP discovery sources:
  ATM2/1/0.1, Src IP addr: dd.dd.dd.dd

```

Check the forwarding table (prefixes, next-hops, and interfaces):

```
Router# show ip cef
```

```

Prefix          Next Hop          Interface
0.0.0.0/0       drop              Null0 (default route handler entry)
0.0.0.0/32      receive
dd.dd.dd.dd/32  dd.dd.dd.dd      ATM2/1/0.1
ee.aa.bb.cc/32  receive
224.0.0.0/4     drop
224.0.0.0/24    receive
255.255.255.255/32 receive

```




---

**Note** Also see the “CSC-PE2 Router Verification: Examples” section on page 75.

---

Verify that interfaces are configured to use LDP:

```
Router# show mpls interfaces
```

```

Interface          IP          Tunnel  Operational
Ethernet0/1        Yes (ldp)   No      Yes

```

Display the entire routing table, including host IP address, next hop, interface, and so forth:

```
Router# show ip route
```

```

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, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR

```

```
Gateway of last resort is not set
```

```

O      dd.0.0.0/32 is subnetted, 1 subnets
      dd.dd.dd.dd [110/7] via dd.dd.dd.dd, 00:16:42, ATM2/1/0.1
C      ee.0.0.0/32 is subnetted, 1 subnets
      ee.ee.ee.ee is directly connected, Loopback0

```

## Configuring VRFs for CSC-PE Routers: Example

The following example shows how to configure a VPN routing/forwarding instance (VRF) for a CSC-PE router:

```

ip cef distributed

ip vrf vpn1
rd 100:1
route target both 100:1
!

```

## Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier: Example

The following example shows how to configure Multiprotocol BGP for VPN connectivity in the backbone carrier:

```

ip cef distributed

ip vrf vpn1
rd 100:1
route target both 100:1

hostname csc-pe1
!
router bgp 100
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 10 30
neighbor ee.ee.ee.ee remote-as 100
neighbor ee.ee.ee.ee update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor ee.ee.ee.ee activate
neighbor ee.ee.ee.ee send-community extended
bgp dampening 30
exit-address-family
!
router bgp 100
. . .
! (BGP IPv4 to CSC-CE router from CSC-PE router)
!
address-family ipv4 vrf vpn1
neighbor ss.0.0.2 remote-as 200
neighbor ss.0.0.2 activate
neighbor ss.0.0.2 as-override
neighbor ss.0.0.2 advertisement-interval 5
neighbor ss.0.0.2 send-label
no auto-summary
no synchronization
bgp dampening 30
exit-address-family
!

```

## Configuring and Verifying the Links Between CSC-PE and CSC-CE Routers: Examples

This section contains the following examples:

- [Configuring EBGP Link with send-label Option for Carrier Supporting Carrier on CSC-PE Routers: Examples, page 68](#)
- [Configuring EBGP Link with send-label Option for Carrier Supporting Carrier on CSC-CE Routers: Examples, page 70](#)
- [Verifying Labels in the CSC-PE Routers: Examples, page 72](#)
- [Verifying Labels in the CSC-CE Routers: Examples, page 78](#)
- [Configuring Route Maps on the CSC-PE Routers: Example, page 84](#)

## Configuring EBGP Link with send-label Option for Carrier Supporting Carrier on CSC-PE Routers: Examples

CSC-PE configuration examples in this section include:

- [CSC-PE1 Configuration: Example, page 68](#)
- [CSC-PE2 Configuration with Multipath Support: Example, page 69](#)

### CSC-PE1 Configuration: Example

The following example shows how to configure a CSC-PE1 router:

```
ip cef
!
ip vrf vpn1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
mpls label protocol ldp
!
interface Loopback0
  ip address dd.dd.dd.dd 255.255.255.255
!
interface Ethernet3/1
  ip vrf forwarding vpn1
  ip address pp.0.0.2 255.0.0.0
!
interface ATM0/1/0
  no ip address
  no ip directed-broadcast
  no ip route-cache distributed
  atm clock INTERNAL
  no atm enable-ilmi-trap
  no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
  ip unnumbered Loopback0
  no ip directed-broadcast
  no atm enable-ilmi-trap
  mpls label protocol ldp
  mpls atm vpi 2-5
  mpls ip
!
```

```

router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/1
 network dd.dd.dd.dd 0.0.0.0 area 100
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
!
address-family vpnv4                                     !VPNv4 session with CSC-PE2
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
!
address-family ipv4 vrf vpn1
 neighbor pp.0.0.1 remote-as 200
 neighbor pp.0.0.1 activate
 neighbor pp.0.0.1 as-override
 neighbor pp.0.0.1 advertisement-interval 5
 neighbor pp.0.0.1 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family

```

### CSC-PE2 Configuration with Multipath Support: Example

The following example shows how to configure a CSC-PE2 router with multipath support:

```

ip cef
!
ip vrf vpn1
 rd 100:1
  route-target export 100:1
  route-target import 100:1
mpls label protocol ldp
!
interface Loopback0
 ip address ee.ee.ee.ee 255.255.255.255
!
interface Ethernet5/0
 ip vrf forwarding vpn1
 ip address ss.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip route-cache distributed
 clock source internal
!
interface Ethernet6/0
 ip vrf forwarding vpn1
 ip address kk.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip route-cache distributed
 clock source internal
!
interface ATM2/1/0
 no ip address
 no ip directed-broadcast

```

```

no ip route-cache distributed
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM2/1/0.1 mpls
ip unnumbered Loopback0
no ip directed-broadcast
no atm enable-ilmi-trap
mpls label protocol ldp
mpls atm vpi 2-5
mpls ip
!
router ospf 100
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets
passive-interface Ethernet5/0
passive-interface ATM3/0/0
network ee.ee.ee.ee 0.0.0.0 area 100
!
router bgp 100
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 10 30
neighbor dd.dd.dd.dd remote-as 100
neighbor dd.dd.dd.dd update-source Loopback0
!
address-family vpnv4                                     !VPNv4 session with CSC-PE1
neighbor dd.dd.dd.dd activate
neighbor dd.dd.dd.dd send-community extended
bgp dampening 30
exit-address-family
!
address-family ipv4 vrf vpn1
maximum-paths 4
neighbor ss.0.0.2 remote-as 200
neighbor ss.0.0.2 activate
neighbor ss.0.0.2 as-override
neighbor ss.0.0.2 advertisement-interval 5
neighbor ss.0.0.2 send-label
neighbor kk.0.0.2 remote-as 200
neighbor kk.0.0.2 activate
neighbor kk.0.0.2 as-override
neighbor kk.0.0.2 advertisement-interval 5
neighbor kk.0.0.2 send-label
no auto-summary
no synchronization
bgp dampening 30
exit-address-family

```

## Configuring EBGW Link with send-label Option for Carrier Supporting Carrier on CSC-CE Routers: Examples

CSC-CE router configuration examples in this section include:

- [CSC-CE1 Configuration: Example, page 71](#)
- [CSC-CE2 Configuration with Multipath Support: Example, page 71](#)

## CSC-CE1 Configuration: Example

The following example shows how to configure a CSC-CE1 router:

```

ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface Ethernet3/0
 ip address pp.0.0.1 255.0.0.0
!
interface Ethernet4/0
 ip address nn.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected                !Exchange routes
 redistribute bgp 200 metric 3 subnets !learned from PE1
 passive-interface ATM1/0
 passive-interface Ethernet3/0
 network cc.cc.cc.cc 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor pp.0.0.2 remote-as 100
 neighbor pp.0.0.2 update-source Ethernet3/0
 no auto-summary
!
address-family ipv4
 redistribute connected
 redistribute ospf 200 metric 4 match internal
 neighbor pp.0.0.2 activate
 neighbor pp.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family

```

## CSC-CE2 Configuration with Multipath Support: Example

This example shows how to configure a CSC-CE2 router with multipath support:

```

ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address gg.gg.gg.gg 255.255.255.255
!
interface Ethernet0/0/3
 ip address ss.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache

```

```

mpls label protocol ldp
mpls ip
!
interface Ethernet6/0
ip address kk.0.0.2 255.0.0.0
no ip directed-broadcast
no atm enable-ilmi-trap
no ip mroute-cache
!
interface ATM3/1/0.1 point2point
ip address yy.0.0.1 255.0.0.0
no ip directed-broadcast
no atm enable-ilmi-trap
no ip mroute-cache
!
router ospf 200
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets          !Exchange routes
redistribute bgp 200 metric 3 subnets    !learned from PE2.
passive-interface ATM3/1/0.1
network gg.gg.gg.gg 0.0.0.0 area 200
network yy.0.0.0 0.255.255.255 area 200
!
router bgp 200
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 10 30
no auto-summary
!
address-family ipv4
maximum-paths 4
redistribute connected
redistribute ospf 200 metric 4 match internal
neighbor ss.0.0.1 activate
neighbor ss.0.0.1 send-label
neighbor kk.0.0.1 activate
neighbor kk.0.0.1 send-label
no auto-summary
no synchronization
bgp dampening 30
exit-address-family

```

## Verifying Labels in the CSC-PE Routers: Examples

Commands that verify labels and their output on the CSC-PE router included in this section are as follows:

- [CSC-PE1 Router Verification: Examples, page 72](#)
- [CSC-PE2 Router Verification: Examples, page 75](#)

### CSC-PE1 Router Verification: Examples

Verify that the BGP session is up and running between the CSC-PE1 router and the CSC-CE1 router. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

```

Router# show ip bgp vpnv4 all summary

BBGP router identifier dd.dd.dd.dd, local AS number 100
BGP table version is 52, main routing table version 52
12 network entries and 13 paths using 2232 bytes of memory

```



```

6 BGP path attribute entries using 336 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 16/4 prefixes, 27/14 paths, scan interval 5 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
ee.ee.ee.ee	4	100	7685	7686	52	0	0	21:17:04	6
pp.0.0.2	4	200	7676	7678	52	0	0	21:16:43	7

Verify that the MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. LDP is turned off on the VRF because EBGp distributes the labels.

```
Router# show mpls interfaces all
```

Interface	IP	Tunnel	Operational
GigabitEthernet6/0	Yes (ldp)	No	Yes

```
VRF vpn1:
```

Interface	IP	Tunnel	Operational
Ethernet3/1	No	No	Yes

Verify that the prefix for the PE1 router is in the routing table of the CSC-PE1 router:

```
Router# show ip route vrf vpn2 bb.bb.bb.bb
```

```

Routing entry for bb.bb.bb.bb/32
  Known via "bgp 100", distance 20, metric 4
  Tag 200, type external
  Last update from pp.0.0.2 21:28:39 ago
  Routing Descriptor Blocks:
  * pp.0.0.2, from pp.0.0.2, 21:28:39 ago
    Route metric is 4, traffic share count is 1
    AS Hops 1, BGP network version 0

```

Verify that the prefix for the PE2 router is in the routing table of the CSC-PE1 router:

```
Router# show ip route vrf vpn2 hh.hh.hh.hh
```

```

Routing entry for hh.hh.hh.hh/32
  Known via "bgp 100", distance 200, metric 4
  Tag 200, type internal
  Last update from ee.ee.ee.ee 21:27:39 ago
  Routing Descriptor Blocks:
  * ee.ee.ee.ee (Default-IP-Routing-Table), from ee.ee.ee.ee, 21:27:39 ago
    Route metric is 4, traffic share count is 1
    AS Hops 1, BGP network version 0

```

Verify that the prefixes for the customer carrier MPLS VPN service provider networks are in the BGP table, and have appropriate labels:

```
Router# show ip bgp vpnv4 vrf vpn2 labels
```

Network	Next Hop	In label/Out label
Route Distinguisher: 100:1 (vpn1)		
cc.cc.cc.cc/32	pp.0.0.2	22/imp-null
bb.bb.bb.bb/32	pp.0.0.2	27/20
hh.hh.hh.hh/32	ee.ee.ee.ee	34/35
gg.gg.gg.gg/32	ee.ee.ee.ee	30/30
nn.0.0.0	pp.0.0.2	23/imp-null
ss.0.0.0	ee.ee.ee.ee	33/34

```
pp.0.0.0          pp.0.0.2          25/aggregate(vpn1)
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider (PE1) is in the CEF table:

```
Router# show ip cef vrf vpn2 bb.bb.bb.bb
bb.bb.bb.bb/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 27
  fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
via pp.0.0.2, 0 dependencies, recursive
next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
valid cached adjacency
tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
```

```
Router# show ip cef vrf vpn2 bb.bb.bb.bb detail
```

```
bb.bb.bb.bb/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 27
  fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
via pp.0.0.2, 0 dependencies, recursive
next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
valid cached adjacency
tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider (PE1) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 bb.bb.bb.bb
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
27	20	bb.bb.bb.bb/32 [V]	958048	Et3/1	pp.0.0.2

```
Router# show mpls forwarding-table vrf vpn2 bb.bb.bb.bb detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
27	20	bb.bb.bb.bb/32 [V]	958125	Et3/1	pp.0.0.2

MAC/Encaps=14/18, MTU=1500, Tag Stack{20}  
00B04A74A05400B0C26E10558847 00014000  
VPN route: vpn1  
No output feature configured  
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider (PE2) is in the CEF table:

```
Router# show ip cef vrf vpn2 hh.hh.hh.hh
```

```
hh.hh.hh.hh/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 34
  fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
via ee.ee.ee.ee, 0 dependencies, recursive
next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
valid cached adjacency
tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
```

```
Router# show ip cef vrf vpn2 hh.hh.hh.hh detail
```

```
hh.hh.hh.hh/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 34
  fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
via ee.aa.aa.aa, 0 dependencies, recursive
  next hop rr.0.0.2, GigabitEthernet6/0 via ee.aa.aa.aa/32
  valid cached adjacency
  tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider (PE2 router) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 hh.hh.hh.hh
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
34	35	hh.hh.hh.hh/32 [V]	139034	Gi6/0	rr.0.0.2

```
Router# show mpls forwarding-table vrf vpn2 hh.hh.hh.hh detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
34	35	hh.hh.hh.hh/32 [V]	139034	Gi6/0	rr.0.0.2

MAC/Encaps=14/18, MTU=1500, Tag Stack{35}  
00B0C26E447000B0C26E10A88847 00023000  
VPN route: vpn1  
No output feature configured  
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

## CSC-PE2 Router Verification: Examples

Verify that the BGP session is up and running between the CSC-PE2 router and the CSC-CE2 router. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

```
Router# show ip bgp vpnv4 all summary
```

```
BGP router identifier ee.aa.aa.aa, local AS number 100
BGP table version is 51, main routing table version 51
12 network entries and 13 paths using 2232 bytes of memory
6 BGP path attribute entries using 336 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 16/4 prefixes, 31/18 paths, scan interval 5 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
dd.dd.dd.dd	4	100	7901	7900	51	0	0	21:52:59	7
ss.0.0.2	4	200	7871	7880	51	0	0	21:50:15	6
kk.0.0.2	4	200	7871	7880	51	0	0	21:50:17	6

Verify that the MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. LDP is turned off on the VRF because EBGp distributes the labels.

```
Router# show mpls interfaces all
```

Interface	IP	Tunnel	Operational
GigabitEthernet4/0	Yes (ldp)	No	Yes

```
VRF vpn1:
Ethernet5/0          No          No          Yes
Ethernet6/0          No          No          Yes
```

Verify that the prefix of the PE1 router is in the routing table of the CSC-PE2 router:

```
Router# show ip route vrf vpn2 bb.bb.bb.bb.bb

Routing entry for bb.bb.bb.bb/32
  Known via "bgp 100", distance 200, metric 4
  Tag 200, type internal
  Last update from dd.dd.dd.dd 21:53:30 ago
  Routing Descriptor Blocks:
  * dd.dd.dd.dd (Default-IP-Routing-Table), from dd.dd.dd.dd, 21:53:30 ago
    Route metric is 4, traffic share count is 1
    AS Hops 1, BGP network version 0
```

Verify that the prefix of the PE2 router is in the routing table of the CSC-PE2 router:

```
Router# show ip route vrf vpn2 hh.hh.hh.hh

Routing entry for hh.hh.hh.hh/32
  Known via "bgp 100", distance 20, metric 4
  Tag 200, type external
  Last update from ss.0.0.2 21:53:12 ago
  Routing Descriptor Blocks:
  * ss.0.0.2, from ss.0.0.2, 21:53:12 ago
    Route metric is 4, traffic share count is 1
    AS Hops 1, BGP network version 0
  kk.0.0.2, from kk.0.0.2, 21:53:12 ago
    Route metric is 4, traffic share count is 1
    AS Hops 1, BGP network version 0
```

Verify that the prefixes for the customer carrier MPLS VPN service provider networks are in the BGP routing table, and that the prefixes have appropriate labels:

```
Router# show ip bgp vpnv4 vrf vpn2 labels

Network          Next Hop          In label/Out label
Route Distinguisher: 100:1 (vpn1)
  cc.cc.cc.cc/32  dd.dd.dd.dd      27/22
  bb.bb.bb.bb/32  dd.dd.dd.dd      26/27
  hh.hh.hh.hh/32  ss.0.0.2         35/31
                  kk.0.0.2         35/31
  gg.gg.gg.gg/32  ss.0.0.2         30/imp-null
  nn.0.0.0         dd.dd.dd.dd      24/23
  ss.0.0.0         ss.0.0.2         34/aggregate (vpn1)
  pp.0.0.0         dd.dd.dd.dd      21/25
  kk.0.0.0         kk.0.0.2         36/aggregate (vpn1)
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider (PE1) is in the CEF table:

```
Router# show ip cef vrf vpn2 bb.bb.bb.bb

bb.bb.bb.bb/32, version 15, cached adjacency rr.0.0.1
0 packets, 0 bytes
tag information set
  local tag: 26
  fast tag rewrite with Gi4/0, rr.0.0.1, tags imposed {27}
via dd.dd.dd.dd, 0 dependencies, recursive
next hop rr.0.0.1, GigabitEthernet4/0 via dd.dd.dd.dd/32
```

```

valid cached adjacency
tag rewrite with Gi4/0, rr.0.0.1, tags imposed {27}

```

```
Router# show ip cef vrf vpn2 bb.bb.bb.bb detail
```

```

bb.bb.bb.bb/32, version 15, cached adjacency rr.0.0.1
0 packets, 0 bytes
tag information set
local tag: 26
fast tag rewrite with Gi4/0, rr.0.0.1, tags imposed {27}
via dd.dd.dd.dd, 0 dependencies, recursive
next hop rr.0.0.1, GigabitEthernet4/0 via dd.dd.dd.dd/32
valid cached adjacency
tag rewrite with Gi4/0, rr.0.0.1, tags imposed {27}

```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider (PE1) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 bb.bb.bb.bb
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
26	27	bb.bb.bb.bb/32[V]	967450	Gi4/0	rr.0.0.1

```
Router# show mpls forwarding-table vrf vpn2 bb.bb.bb.bb detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
26	27	bb.bb.bb.bb/32[V]	967510	Gi4/0	rr.0.0.1

```

MAC/Encaps=14/18, MTU=1500, Tag Stack{27}
00B0C26E10A800B0C26E44708847 0001B000
VPN route: vpn1
No output feature configured
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider (PE2) is in the CEF table:

```
Router# show ip cef vrf vpn2 hh.hh.hh.hh
```

```

hh.hh.hh.hh/32, version 33, cached adjacency ss.0.0.2
0 packets, 0 bytes
tag information set
local tag: 35
via ss.0.0.2, 0 dependencies, recursive
next hop ss.0.0.2, Ethernet5/0 via ss.0.0.2/32
valid cached adjacency
tag rewrite with Et5/0, ss.0.0.2, tags imposed {31}
via kk.0.0.2, 0 dependencies, recursive
next hop kk.0.0.2, Ethernet6/0 via kk.0.0.2/32
valid cached adjacency
tag rewrite with Et6/0, kk.0.0.2, tags imposed {31}

```

```
Router# show ip cef vrf vpn2 hh.hh.hh.hh detail

hh.hh.hh.hh/32, version 33, cached adjacency ss.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 35
via ss.0.0.2, 0 dependencies, recursive
  next hop ss.0.0.2, Ethernet5/0 via ss.0.0.2/32
  valid cached adjacency
  tag rewrite with Et5/0, ss.0.0.2, tags imposed {31}
via kk.0.0.2, 0 dependencies, recursive
  next hop kk.0.0.2, Ethernet6/0 via kk.0.0.2/32
  valid cached adjacency
  tag rewrite with Et6/0, kk.0.0.2, tags imposed {31}
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider (PE2) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf <CSC-PE-vrf-name> hh.hh.hh.hh
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
35	31	hh.hh.hh.hh/32 [V]	2023332	Et5/0	ss.0.0.2
				Et6/0	kk.0.0.2

```
Router# show mpls forwarding-table vrf <CSC-PE-vrf-name> hh.hh.hh.hh detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
35	31	hh.hh.hh.hh/32 [V]	2023332	Et5/0	ss.0.0.2
				Et6/0	kk.0.0.2

MAC/Encaps=14/18, MTU=1500, Tag Stack{31}  
0030A363380300B0C26E448C8847 0001F000  
VPN route: vpn1  
No output feature configured  
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

## Verifying Labels in the CSC-CE Routers: Examples

Commands that verify labels and their output on the CSC-PE router included in this section are as follows:

- [CSC-CE1 Router Verification: Examples, page 78](#)
- [CSC-CE2 Router Verification: Examples, page 80](#)

### CSC-CE1 Router Verification: Examples

Verify that the BGP session is up and running:

```
Router# show ip bgp summary
```

```
BGP router identifier cc.cc.cc.cc, local AS number 200
BGP table version is 35, main routing table version 35
14 network entries and 14 paths using 2030 bytes of memory
3 BGP path attribute entries using 168 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 1 history paths, 0 dampened paths
BGP activity 17/67 prefixes, 29/15 paths, scan interval 60 secs
```

```
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
pp.0.0.1      4      100   7615   7613     35    0    0 21:06:19      5
```

Verify that the loopback address of the local PE router (PE1) is in the routing table:

```
Router# show ip route bb.bb.bb.bb
```

```
Routing entry for bb.bb.bb.bb/32
  Known via "ospf 200", distance 110, metric 101, type intra area
  Redistributing via bgp 200
  Advertised by bgp 200 metric 4 match internal
  Last update from nn.0.0.1 on Ethernet4/0, 00:34:08 ago
  Routing Descriptor Blocks:
  * nn.0.0.1, from bb.bb.bb.bb, 00:34:08 ago, via Ethernet4/0
    Route metric is 101, traffic share count is 1
```

Verify that the loopback address of the remote PE router (PE2) is in the routing table:

```
Router# show ip route hh.hh.hh.hh
```

```
Routing entry for hh.hh.hh.hh/32
  Known via "bgp 200", distance 20, metric 0
  Tag 100, type external
  Redistributing via ospf 200
  Advertised by ospf 200 metric 3 subnets
  Last update from pp.0.0.1 00:45:16 ago
  Routing Descriptor Blocks:
  * pp.0.0.1, from pp.0.0.1, 00:45:16 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2, BGP network version 0
```

Verify that the prefix of the local PE router (PE1) is in the MPLS LDP bindings:

```
Router# show mpls ldp bindings bb.bb.bb.bb 255.255.255.255
```

```
tib entry: bb.bb.bb.bb/32, rev 20
  local binding: tag: 20
  remote binding: tsr: bb.bb.bb.bb:0, tag: imp-null
```

Verify that the prefix of the local PE router (PE1) is in the CEF table:

```
Router# show ip cef bb.bb.bb.bb
```

```
bb.bb.bb.bb/32, version 46, cached adjacency nn.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 20
  via nn.0.0.1, Ethernet4/0, 0 dependencies
  next hop nn.0.0.1, Ethernet4/0
  unresolved
  valid cached adjacency
  tag rewrite with Et4/0, nn.0.0.1, tags imposed {}
```

Verify that the prefix of the local PE router (PE1) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table bb.bb.bb.bb
```

Local tag	Outgoing tag or VC Pop tag	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
20	Pop tag	bb.bb.bb.bb/32	893397	Et4/0	nn.0.0.1

```
Router# show mpls forwarding-table bb.bb.bb.bb detail
```

Local tag	Outgoing tag	Prefix	Bytes tag	Outgoing interface	Next Hop
20	Pop tag	bb.bb.bb.bb/32	893397	Et4/0	nn.0.0.1

```

tag    tag or VC    or Tunnel Id    switched    interface
20    Pop tag    bb.bb.bb.bb/32    893524    Et4/0    nn.0.0.1
      MAC/Encaps=14/14, MTU=1504, Tag Stack{
      00074F83685400B04A74A0708847
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

Verify that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks:

```
Router# show ip bgp labels
```

```

Network          Next Hop          In Label/Out Label
cc.cc.cc.cc/32   0.0.0.0           imp-null/exp-null
bb.bb.bb.bb/32   nn.0.0.1          20/exp-null
hh.hh.hh.hh/32   pp.0.0.1          26/34
gg.gg.gg.gg/32   pp.0.0.1          23/30
nn.0.0.0         0.0.0.0           imp-null/exp-null
ss.0.0.0         pp.0.0.1          25/33
pp.0.0.0         0.0.0.0           imp-null/exp-null
pp.0.0.1/32     0.0.0.0           16/exp-null

```

Verify that the prefix of the remote PE router (PE2) is in the CEF table:

```
Router# show ip cef hh.hh.hh.hh
```

```

hh.hh.hh.hh/32, version 54, cached adjacency pp.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 26
    fast tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
  via pp.0.0.1, 0 dependencies, recursive
  next hop pp.0.0.1, Ethernet3/0 via pp.0.0.1/32
  valid cached adjacency
  tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}

```

Verify that the prefix of the remote PE router (PE2) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table hh.hh.hh.hh
```

```

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
26     34        hh.hh.hh.hh/32  81786     Et3/0     pp.0.0.1

```

```
Router# show mpls forwarding-table hh.hh.hh.hh detail
```

```

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
26     34        hh.hh.hh.hh/32  81863     Et3/0     pp.0.0.1
      MAC/Encaps=14/18, MTU=1500, Tag Stack{34}
      00B0C26E105500B04A74A0548847 00022000
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

## CSC-CE2 Router Verification: Examples

Verify that the BGP session is up and running:

```
Router# show ip bgp summary
```

```

BGP router identifier gg.gg.gg.gg, local AS number 200
BGP table version is 31, main routing table version 31
13 network entries and 13 paths using 1885 bytes of memory
3 BGP path attribute entries using 168 bytes of memory

```



```

1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 17/4 prefixes, 20/7 paths, scan interval 60 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
ss.0.0.1	4	100	7962	7953	31	0	0	22:03:55	6
kk.0.0.1	4	100	7962	7953	31	0	0	22:03:57	6

Verify that the loopback address of the local PE router (PE2) is in the routing table:

```
Router# show ip route hh.hh.hh.hh
```

```

Routing entry for hh.hh.hh.hh/32
  Known via "ospf 200", distance 110, metric 7, type intra area
  Redistributing via bgp 200
  Advertised by bgp 200 metric 4 match internal
  Last update from 19.19.19.19 on ATM3/1/0.1, 01:37:02 ago
  Routing Descriptor Blocks:
  * hh.hh.hh.hh, from hh.hh.hh.hh, 01:37:02 ago, via ATM3/1/0.1
    Route metric is 7, traffic share count is 1

```

Verify that the loopback address of the remote PE router (PE1) is in the routing table:

```
Router# show ip route bb.bb.bb.bb
```

```

Routing entry for bb.bb.bb.bb/32
  Known via "bgp 200", distance 20, metric 0
  Tag 100, type external
  Redistributing via ospf 200
  Advertised by ospf 200 metric 3 subnets
  Last update from ss.0.0.1 22:04:44 ago
  Routing Descriptor Blocks:
  * ss.0.0.1, from ss.0.0.1, 22:04:44 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2, BGP network version 0
  kk.0.0.1, from .0.0.1, 22:04:44 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2, BGP network version 0

```

Verify that the prefix of the local PE router (PE2) is in the MPLS LDP bindings:

```
Router# show mpls ldp bindings hh.hh.hh.hh 255.255.255.255
```

```

tib entry: hh.hh.hh.hh/32, rev 37
  local binding: tag: 31

```

```
Router# show mpls ldp bindings hh.hh.hh.hh 255.255.255.255 detail
```

```

tib entry: hh.hh.hh.hh/32, rev 37
  local binding: tag: 31

```

Verify that the prefix of the local PE (PE2) is in the CEF table:

```
Router# show ip cef hh.hh.hh.hh
```

```

hh.hh.hh.hh/32, version 31, cached adjacency to ATM3/1/0.1
0 packets, 0 bytes
tag information set
  local tag: 31
  fast tag rewrite with ATM3/1/0.1, point2point, tags imposed {2/33(vcd=2)}
via hh.hh.hh.hh, ATM3/1/0.1, 0 dependencies
next hop hh.hh.hh.hh, ATM3/1/0.1

```

```

unresolved
valid cached adjacency
tag rewrite with AT3/1/0.1, point2point, tags imposed {2/33(vcd=2)}

```

Router# **show ip cef hh.hh.hh.hh detail**

```

hh.hh.hh.hh/32, version 31, cached adjacency to ATM3/1/0.1
0 packets, 0 bytes
tag information set
  local tag: 31
  fast tag rewrite with AT3/1/0.1, point2point, tags imposed {2/33(vcd=2)}
via hh.hh.hh.hh, ATM3/1/0.1, 0 dependencies
  next hop hh.hh.hh.hh, ATM3/1/0.1
  unresolved
  valid cached adjacency
  tag rewrite with AT3/1/0.1, point2point, tags imposed {2/33(vcd=2)}

```

Verify that the prefix of the local PE router (PE2) is in the MPLS forwarding table:

Router# **show mpls forwarding-table hh.hh.hh.hh**

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
31	2/33	hh.hh.hh.hh/32	1908083	AT3/1/0.1	point2point

Router# **show mpls forwarding-table hh.hh.hh.hh detail**

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
31	2/33	hh.hh.hh.hh/32	1908200	AT3/1/0.1	point2point
MAC/Encaps=4/8, MTU=4470, Tag Stack{2/33(vcd=2)}					
00028847 00002000					
No output feature configured					
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15					

Verify that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks:

Router# **show ip bgp labels**

Network	Next Hop	In Label/Out Label
cc.cc.cc.cc/32	ss.0.0.1	18/27
	kk.0.0.1	18/27
bb.bb.bb.bb/32	ss.0.0.1	19/26
	kk.0.0.1	19/26
hh.hh.hh.hh/32	hh.hh.hh.hh	31/exp-null
gg.gg.gg.gg/32	0.0.0.0	imp-null/exp-null
nn.0.0.0	ss.0.0.1	22/24
	kk.0.0.1	22/24
ss.0.0.0	0.0.0.0	imp-null/exp-null
ss.0.0.1/32	0.0.0.0	16/exp-null
pp.0.0.0	ss.0.0.1	26/21
	kk.0.0.1	26/21
kk.0.0.1/32	0.0.0.0	17/exp-null

Verify that the prefix of the remote PE router (PE1) is in the CEF table:

```
Router# show ip cef bb.bb.bb.bb
```

```
bb.bb.bb.bb/32, version 18, cached adjacency ss.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 19
    fast tag rewrite with Et0/0/3, ss.0.0.1, tags imposed {26}
  via ss.0.0.1, 0 dependencies, recursive
    next hop ss.0.0.1, Ethernet0/0/3 via ss.0.0.1/32
    valid cached adjacency
    tag rewrite with Et0/0/3, ss.0.0.1, tags imposed {26}
  via kk.0.0.1, 0 dependencies, recursive
    next hop kk.0.0.1, Ethernet6/0 via kk.0.0.1/32
    valid cached adjacency
    tag rewrite with Et6/0, kk.0.0.1, tags imposed {26}
```

```
Router# show ip cef bb.bb.bb.bb detail
```

```
bb.bb.bb.bb/32, version 18, cached adjacency ss.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 19
    fast tag rewrite with Et0/0/3, ss.0.0.1, tags imposed {26}
  via ss.0.0.1, 0 dependencies, recursive
    next hop ss.0.0.1, Ethernet0/0/3 via ss.0.0.1/32
    valid cached adjacency
    tag rewrite with Et0/0/3, ss.0.0.1, tags imposed {26}
  via kk.0.0.1, 0 dependencies, recursive
    next hop kk.0.0.1, Ethernet6/0 via kk.0.0.1/32
    valid cached adjacency
    tag rewrite with Et6/0, kk.0.0.1, tags imposed {26}
```

Verify that the prefix of the remote PE router (PE1) is in the MPLS forwarding table:

```
Router# show mpls forwarding-table bb.bb.bb.bb
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
19	26	bb.bb.bb.bb/32	965401	Et0/0/3	ss.0.0.1
	26	bb.bb.bb.bb/32	156001	Et6/0	kk.0.0.1

```
Router# show mpls forwarding-table bb.bb.bb.bb detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
19	26	bb.bb.bb.bb/32	965478	Et0/0/3	ss.0.0.1
MAC/Encaps=14/18, MTU=1500, Tag Stack{26}					
00B0C26E448C0030A36338038847 0001A000					
No output feature configured					
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15					
	26	bb.bb.bb.bb/32	965478	Et6/0	kk.0.0.1
MAC/Encaps=14/18, MTU=1500, Tag Stack{26}					
00B0C26E448C2480A36338039974 0001A000					
No output feature configured					
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15					

## Configuring Route Maps on the CSC-PE Routers: Example

The following example creates two route maps, which are named:

- IN for incoming routes
- OUT for outgoing routes

The route maps specify the following:

- If an IP address in an incoming BGP update message matches an IP address in access list 99, the route is added to the BGP table.
- If an IP address in an outbound BGP update message matches an IP address in access list 88, the router distributes that route.

The route maps are applied to the CSC-PE router with the address qq.0.0.1.

```
address-family ipv4 vrf vpn2
 neighbor qq.0.0.1 remote-as 200
 neighbor qq.0.0.1 activate
 neighbor qq.0.0.1 as-override
 neighbor qq.0.0.1 advertisement-interval 5
 neighbor qq.0.0.1 route-map IN in
 neighbor qq.0.0.1 route-map OUT out
 neighbor qq.0.0.1 send-label
!
access-list 88 permit rr.rr.rr.rr
access-list 88 permit ss.ss.ss.ss
access-list 88 permit tt.tt.tt.tt
access-list 99 permit uu.uu.uu.uu
access-list 99 permit vv.vv.vv.vv
access-list 99 permit ww.ww.ww.ww
!
route-map IN permit 1
 match ip address 99
!
route-map OUT permit 1
 match ip address 88
 set mpls-label
!
```

## Configuring and Verifying the Customer Carrier Network: Examples

Customer carrier configuration and verification examples in this section include:

- [Verifying IP Connectivity in the Customer Carrier: Example, page 84](#)
- [Configuring a Customer Carrier Core Router as a Route Reflector: Example, page 85](#)

### Verifying IP Connectivity in the Customer Carrier: Example

Verify the connectivity from one customer carrier core router to another (from CE1 to CE2) by entering the following command:

```
Router# ping jj.jj.jj.jj

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to jj.jj.jj.jj, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
```

Verify the path that a packet goes through on its way to its final destination from CE1 to CE2:

```
Router# trace jj.jj.jj.jj
```

```
Type escape sequence to abort.
Tracing the route to jj.jj.jj.jj
```

```
 1 mm.0.0.2 0 msec 0 msec 4 msec
 2 nn.0.0.2 [MPLS: Labels 20/21 Exp 0] 8 msec 8 msec 12 msec
 3 pp.0.0.2 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 12 msec
 4 ss.0.0.1 [MPLS: Labels 17/21 Exp 0] 8 msec 8 msec 12 msec
 5 ss.0.0.2 [MPLS: Labels 16/21 Exp 0] 8 msec 8 msec 12 msec
 6 tt.0.0.1 [AS 200] [MPLS: Label 21 Exp 0] 8 msec 8 msec 8 msec
 7 tt.0.0.2 [AS 200] 8 msec 4 msec *
```

Verify the path that a packet goes through on its way to its final destination from CE2 to CE1:

```
Router# trace aa.aa.aa.aa
```

```
Type escape sequence to abort.
Tracing the route to aa.aa.aa.aa
```

```
 1 tt.0.0.1 0 msec 0 msec 0 msec
 2 qq.0.0.2 [MPLS: Labels 18/21 Exp 0] 8 msec 12 msec 12 msec
 3 ss.0.0.1 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 8 msec
 4 pp.0.0.2 [MPLS: Labels 17/21 Exp 0] 12 msec 8 msec 8 msec
 5 pp.0.0.1 [MPLS: Labels 16/21 Exp 0] 12 msec 12 msec 8 msec
 6 mm.0.0.2 [AS 200] [MPLS: Label 21 Exp 0] 12 msec 8 msec 12 msec
 7 mm.0.0.1 [AS 200] 4 msec 4 msec *
```

## Configuring a Customer Carrier Core Router as a Route Reflector: Example

The following example shows how to use an address family to configure internal BGP peer 10.1.1.1 as a route-reflector client for both unicast and multicast prefixes:

```
router bgp 200
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client

router bgp 100
 address-family vpnv4
  neighbor xx.xx.xx.xx activate
  neighbor xx.xx.xx.xx route-reflector-client
  ! xx.xx.xx,xx is a PE router
  neighbor xx.xx.xx.xx send-community extended
 exit address-family
! You need to configure your peer BGP neighbor.
```

## Configuring and Verifying the Customer Site for Hierarchical VPNs: Examples

This section contains the following configuration and verification examples for the customer site:

- [Configuring Provider Edge Routers for Hierarchical VPNs: Examples, page 86](#)
- [Verifying Labels in Each Provider Edge Router for Hierarchical VPNs: Examples, page 88](#)
- [Configuring Customer Edge Routers for Hierarchical VPNs: Examples, page 92](#)
- [Verifying IP Connectivity in the Customer Site: Examples, page 93](#)

### Configuring Provider Edge Routers for Hierarchical VPNs: Examples

Provider edge (PE) router configuration examples in this section include:

- [PE1 Router Configuration: Example, page 86](#)
- [PE2 Router Configuration: Example, page 87](#)

#### PE1 Router Configuration: Example

This example shows how to configure a PE1 router:

```
ip cef
!
ip vrf vpn2
 rd 200:1
  route-target export 200:1
  route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Ethernet3/0
 ip address nn.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/3
 ip vrf forwarding vpn2
 ip address mm.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/3
 network bb.bb.bb.bb 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor hh.hh.hh.hh remote-as 200
 neighbor hh.hh.hh.hh update-source Loopback0
!
```

```

address-family vpnv4                                     !VPNv4 session with PE2
 neighbor hh.hh.hh.hh activate
 neighbor hh.hh.hh.hh send-community extended
 bgp dampening 30
 exit-address-family
!
address-family ipv4 vrf vpn2
 neighbor mm.0.0.1 remote-as 300
 neighbor mm.0.0.1 activate
 neighbor mm.0.0.1 as-override
 neighbor mm.0.0.1 advertisement-interval 5
no auto-summary
no synchronization
 bgp dampening 30
 exit-address-family

```

## PE2 Router Configuration: Example

This example shows how to configure a PE2 router:

```

ip cef
!
ip vrf vpn2
 rd 200:1
  route-target export 200:1
  route-target import 200:1
!
mpls label protocol ldp
!
interface Loopback0
 ip address hh.hh.hh.hh 255.255.255.255
!
interface Ethernet3/6
 ip vrf forwarding vpn2
 ip address tt.0.0.2 255.0.0.0
!
interface ATM5/0.1 point2point
 ip address qq.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
!
address-family vpnv4                                     !VPNv4 session with PE1
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 bgp dampening 30
 exit-address-family
!
address-family ipv4 vrf vpn2
 neighbor tt.0.0.1 remote-as 300
 neighbor tt.0.0.1 activate
 neighbor tt.0.0.1 as-override
 neighbor tt.0.0.1 advertisement-interval 5
no auto-summary

```

```
no synchronization
bgp dampening 30
exit-address-family
```

## Verifying Labels in Each Provider Edge Router for Hierarchical VPNs: Examples

Provider edge (PE) router label verification examples in this section include the following:

- [PE1 Router Label Verification: Examples, page 88](#)
- [PE2 Router Label Verification: Examples, page 89](#)

### PE1 Router Label Verification: Examples

Verify that the loopback address of the local CE router (CE1) is in the routing table of the PE1 router:

```
Router# show ip route vrf vpn2 aa.aa.aa.aa

Routing entry for aa.aa.aa.aa/32
  Known via "bgp 200", distance 20, metric 0
  Tag 300, type external
  Last update from mm.0.0.2 20:36:59 ago
  Routing Descriptor Blocks:
  * mm.0.0.2, from mm.0.0.2, 20:36:59 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1, BGP network version 0
```

Verify that the prefix for the local CE router (CE1) is in the MPLS forwarding table, and that the prefix is untagged:

```
Router# show mpls forwarding-table vrf vpn2 aa.aa.aa.aa

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
23     Untagged  aa.aa.aa.aa/32[V]  0         Et3/3     mm.0.0.2
```

Verify that the prefix of the remote PE router (PE2) is in the Cisco Express Forwarding (CEF) table:

```
Router# show ip cef hh.hh.hh.hh

hh.hh.hh.hh/32, version 31, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 31
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
  via nn.0.0.2, Ethernet3/0, 2 dependencies
    next hop nn.0.0.2, Ethernet3/0
  unresolved
  valid cached adjacency
  tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
```

Verify that the loopback address of the remote CE router (CE2) is in the routing table:

```
Router# show ip route vrf vpn2 jj.jj.jj.jj

Routing entry for jj.jj.jj.jj/32
  Known via "bgp 200", distance 200, metric 0
  Tag 300, type internal
  Last update from hh.hh.hh.hh 20:38:49 ago
  Routing Descriptor Blocks:
  * hh.hh.hh.hh (Default-IP-Routing-Table), from hh.hh.hh.hh, 20:38:49 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1, BGP network version 0
```



Verify that the prefix of the remote CE router (CE2) is in the MPLS forwarding table, and that an outgoing interface exists:

```
Router# show mpls forwarding-table vrf vpn2 jj.jj.jj.jj

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
None   26         jj.jj.jj.jj/32  0         Et3/0     nn.0.0.2
```

Verify that the prefix of the remote CE router (CE2) is in the CEF table:

```
Router# show ip cef vrf vpn2 jj.jj.jj.jj

jj.jj.jj.jj/32, version 12, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
  via hh.hh.hh.hh, 0 dependencies, recursive
    next hop nn.0.0.2, Ethernet3/0 via hh.hh.hh.hh/32
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
```

Verify that the prefix of the local PE router (PE1) is in the CEF table:

```
Router# show ip cef bb.bb.bb.bb

bb.bb.bb.bb/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

## PE2 Router Label Verification: Examples

Verify that the loopback address of the local CE router (CE2) is in the routing table of the PE2 router:

```
Router# show ip route vrf vpn2 jj.jj.jj.jj

Routing entry for jj.jj.jj.jj/32
  Known via "bgp 200", distance 20, metric 0
  Tag 300, type external
  Last update from tt.0.0.2 22:11:06 ago
  Routing Descriptor Blocks:
  * tt.0.0.2, from tt.0.0.2, 22:11:06 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1, BGP network version 0
```

Verify that the prefix of the local CE router (CE2) is in the MPLS forwarding table, and is untagged:

```
Router# show mpls forwarding-table vrf vpn2 jj.jj.jj.jj

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
32     Untagged  jj.jj.jj.jj/32[V]  0         Et3/6     tt.0.0.2
```

```
Router# show mpls forwarding-table vrf vpn2 jj.jj.jj.jj detail

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
32     Untagged  jj.jj.jj.jj/32[V] 0           Et3/6      tt.0.0.2
      MAC/Encaps=0/0, MTU=1504, Tag Stack{}
      VPN route: vpn2
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the prefix of the remote PE router (PE1) is in the CEF table:

```
Router# show ip cef bb.bb.bb.bb

bb.bb.bb.bb/32, version 19, cached adjacency to ATM5/0.1
0 packets, 0 bytes
tag information set
  local tag: 20
  fast tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6)}
via gg.gg.gg.gg, ATM5/0.1, 2 dependencies
  next hop gg.gg.gg.gg, ATM5/0.1
  unresolved
  valid cached adjacency
  tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6)}
```

```
Router# show ip cef bb.bb.bb.bb detail

bb.bb.bb.bb/32, version 19, cached adjacency to ATM5/0.1
0 packets, 0 bytes
tag information set
  local tag: 20
  fast tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6)}
via gg.gg.gg.gg, ATM5/0.1, 2 dependencies
  next hop gg.gg.gg.gg, ATM5/0.1
  unresolved
  valid cached adjacency
  tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6)}
```

Verify that the loopback address of the remote CE router (CE1) is in the routing table:

```
Router# show ip route vrf vpn2 aa.aa.aa.aa

Routing entry for aa.aa.aa.aa/32
  Known via "bgp 200", distance 200, metric 0
  Tag 300, type internal
  Last update from bb.bb.bb.bb 01:43:34 ago
  Routing Descriptor Blocks:
  * bb.bb.bb.bb (Default-IP-Routing-Table), from bb.bb.bb.bb, 01:43:34 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1, BGP network version 0
```

Verify that the prefix of the remote CE router (CE1) is in the MPLS forwarding table, and that it has an outgoing interface:

```
Router# show mpls forwarding-table vrf vpn2 aa.aa.aa.aa

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
None   2/35      aa.aa.aa.aa/32 0           AT5/0.1   point2point
```

```
Router# show mpls forwarding-table vrf vpn2 aa.aa.aa.aa detail
```

```
Local   Outgoing   Prefix           Bytes tag   Outgoing   Next Hop
tag     tag or VC  or Tunnel Id    switched   interface
None    2/35       aa.aa.aa.aa/32  0          AT5/0.1    point2point
        MAC/Encaps=4/12, MTU=4466, Tag Stack{2/35(vcd=6) 23}
        00068847 0000600000017000
        No output feature configured
```

Verify that the prefix of the remote CE router (CE1) is in the CEF table:

```
Router# show ip cef vrf vpn2 aa.aa.aa.aa
```

```
aa.aa.aa.aa/32, version 10, cached adjacency to ATM5/0.1
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6) 23}
  via bb.bb.bb.bb, 0 dependencies, recursive
  next hop gg.gg.gg.gg, ATM5/0.1 via bb.bb.bb.bb/32
  valid cached adjacency
  tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6) 23}
```

```
Router# show ip cef vrf vpn2 aa.aa.aa.aa detail
```

```
aa.aa.aa.aa/32, version 10, cached adjacency to ATM5/0.1
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6) 23}
  via bb.bb.bb.bb, 0 dependencies, recursive
  next hop gg.gg.gg.gg, ATM5/0.1 via bb.bb.bb.bb/32
  valid cached adjacency
  tag rewrite with AT5/0.1, point2point, tags imposed {2/35(vcd=6) 23}
```

Verify that the prefix of the local PE router (PE2) is in the CEF table:

```
Router# show ip cef hh.hh.hh.hh
```

```
hh.hh.hh.hh/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

```
Router# show ip cef hh.hh.hh.hh detail
```

```
hh.hh.hh.hh/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

## Configuring Customer Edge Routers for Hierarchical VPNs: Examples

Customer edge (CE) router configuration examples in this section include:

- [CE1 Configuration: Example, page 92](#)
- [CE2 Configuration: Example, page 92](#)

### CE1 Configuration: Example

The following example shows how to configure a CE1 router:

```
ip cef
interface Loopback0
ip address aa.aa.aa.aa 255.255.255.255
!
interface Ethernet3/3
ip address mm.0.0.1 255.0.0.0
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30
redistribute connected                                !Redistributing routes into BGP
neighbor mm.0.0.2 remote-as 200                       !to send to PE1
neighbor mm.0.0.2 advertisement-interval 5
no auto-summary
```

### CE2 Configuration: Example

The following example shows how to configure a CE2 router:

```
ip cef
!
interface Loopback0
ip address jj.jj.jj.jj 255.255.255.255
!
interface Ethernet3/6
ip address tt.0.0.1 255.0.0.0
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30                                !Redistributing static routes into BGP
redistribute connected                            !to send to PE2
redistribute ospf 300 match internal external 1 external 2
neighbor tt.0.0.2 remote-as 200
neighbor tt.0.0.2 advertisement-interval 5
no auto-summary
```

## Verifying IP Connectivity in the Customer Site: Examples

Customer edge (CE) router verification examples in this section include:

- [CE1 Router Verification: Example, page 93](#)
- [CE2 Router Verification: Example, page 93](#)

### CE1 Router Verification: Example

Verify that the loopback address of the remote CE router (CE2), learned from the PE router, is in the routing table of the CE1 router:

```
Router# show ip route jj.jj.jj.jj

Routing entry for jj.jj.jj.jj/32
  Known via "bgp 300", distance 20, metric 0
  Tag 200, type external
  Redistributing via ospf 300
  Advertised by ospf 300 subnets
  Last update from mm.0.0.1 20:29:35 ago
  Routing Descriptor Blocks:
  * mm.0.0.1, from mm.0.0.1, 20:29:35 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2
```

### CE2 Router Verification: Example

Verify that the loopback address of the remote CE router (CE1), learned from the PE router, is in the routing table of the CE2 router:

```
Router# show ip route aa.aa.aa.aa

Routing entry for aa.aa.aa.aa/32
  Known via "bgp 300", distance 20, metric 0
  Tag 200, type external
  Redistributing via ospf 300
  Advertised by ospf 300 subnets
  Last update from tt.0.0.1 22:16:46 ago
  Routing Descriptor Blocks:
  * tt.0.0.1, from tt.0.0.1, 22:16:46 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2
```

## Additional References

The following sections provide references related to MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution.

## Related Documents

Related Topic	Document Title
MPLS configuration tasks	<a href="#">“Configuring Multiprotocol Label Switching” chapter in the <i>Cisco IOS Switching Services Configuration Guide, Release 12.0</i></a>
MPLS VPN configuration tasks	<a href="#">MPLS Virtual Private Networks (VPNs)</a>
Enhanced MPLS VPN traffic management configuration tasks	<a href="#">MPLS Virtual Private Network Enhancements</a>
Basic MPLS VPN Carrier Supporting Carrier configuration tasks	<a href="#">MPLS VPN Carrier Supporting Carrier</a>
BGP configuration tasks	“Configuring BGP chapter” in the <a href="#">Cisco IOS IP Configuration Guide, Release 12.0</a>
An explanation of how BGP works and how you can use it to participate in routing with other networks that run BGP	<a href="#">Using the Border Gateway Protocol for Interdomain Routing</a>
An explanation of the purpose of the BGP and the BGP route selection process, and how to use BGP attributes in route selection	“Border Gateway Protocol” chapter in the <a href="#">Internetworking Technology Overview</a>
Commands to configure and monitor BGP	“Border Gateway Protocol” chapter in the <a href="#">Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.0</a>
MPLS LDP configuration tasks	<a href="#">MPLS Label Distribution Protocol (LDP)</a>
OSPF configuration tasks	“Configuring OSPF” chapter in the <a href="#">Cisco IOS IP Configuration Guide, Part II: IP Routing Protocols, Release 12.0</a>
IS-IS configuration tasks	“Configuring Integrated IS-IS chapter” in the <a href="#">Cisco IOS IP Configuration Guide, Part II: IP Routing Protocol, Release 12.0</a>
Extended <b>ping</b> and extended <b>tracert</b> commands	<a href="#">Using the Extended <b>ping</b> and Extended <b>tracert</b> Commands</a>
Explicit null labels	MPLS VPN—Explicit Null Label Support with BGP IPv4 Label Session

## Standards

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

## MIBs

MIBs	MIBs Link
No new or modified MIBs are supported 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: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

## RFCs

RFCs	Title
RFC 1164	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1171	<i>A Border Gateway Protocol 4</i>
RFC 1700	<i>Assigned Numbers</i>
RFC 1966	<i>BGP Route Reflection: An Alternative to Full Mesh IBGP</i>
RFC 2283	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2547	<i>BGP/MPLS VPNs</i>
RFC 2842	<i>Capabilities Advertisement with BGP-4</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 3107	<i>Carrying Label Information in BGP-4</i>

## 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.	<a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a>

# Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3 command reference publications.

- [mpls bgp forwarding](#)



# mpls bgp forwarding

To enable an interface to receive Multiprotocol Label Switching (MPLS) packets when the signaling of MPLS labels is through the use of the Border Gateway Protocol (BGP), use the **mpls bgp forwarding** command in interface configuration mode. To disable an interface from receiving MPLS packets when the signaling of MPLS labels is through the use of the BGP, use the **no** form of this command.

**mpls bgp forwarding**

**no mpls bgp forwarding**

**Syntax Description** This command has no arguments or keywords.

**Defaults** MPLS forwarding by BGP is not enabled.

**Command Modes** Interface configuration

Command History	Release	Modification
	12.0(29)S	This command was introduced.

**Usage Guidelines** Use the **mpls bgp forwarding** command when you want to enable MPLS forwarding on directly connected loopback interfaces. This command is automatically generated by BGP for directly connected nonloopback neighbors.

**Examples** The following example shows how to configure BGP to enable MPLS forwarding on a directly connected loopback interface, Ethernet 0/0:

```
Router(config)# interface ethernet 0/0
Router(config-if)# bgp mpls forwarding
```

Related Commands	Command	Description
	<b>ip vrf forwarding</b>	Associates a Virtual Private Network (VPN) routing/forwarding instance (VRF) with an interface or subinterface.

# Glossary

**AS**—autonomous system. A collection of networks that share the same routing protocol and that are under the same system administration.

**BGP**—Border Gateway Protocol. The exterior border gateway protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

**BGP prefix**—A route announcement using the BGP. A prefix is composed of a path of autonomous system numbers, indicating which networks the packet must pass through, and the IP block that is being routed. A BGP prefix would look something like: 701 1239 42 206.24.14.0/24. (The /24 part is referred to as a CIDR mask.) The /24 indicates that there are 24 ones in the netmask for this block starting from the left side. A /24 corresponds to the natural mask 255.255.255.0.

**CE router**—customer edge router. The customer router that connects to the provider edge (PE) router.

**EBGP**—External Border Gateway Protocol. A BGP session between routers in different autonomous systems (ASs). When a pair of routers in different ASs are more than one IP hop away from each other, an external BGP session between those two routers is called multihop external BGP.

**IBGP**—Internal Border Gateway Protocol. A BGP session between routers within the same autonomous system.

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

**LDP**—Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

**LER**—label edge router. The edge router that performs label imposition and disposition.

**LSP**—label-switched path. A sequence of hops in which a packet travels from one router to another router by means of label switching mechanisms. A label-switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

**LSR**—label switch router. A router that forwards packets in an MPLS network by looking only at the fixed-length label.

**MPLS**—Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. MPLS enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels.

**Multihop BGP**—A Border Gateway Protocol between two routers in different autonomous systems that are more than one hop away from each other.

**NLRI**—Network Layer Reachability Information. BGP sends routing update messages containing NLRI, which describes the route. 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.

**P router**—provider router. The core router in the service provider network that connects to provider edge (PE) routers. In a packet-switched star topology, a router that is part of the backbone and that serves as the single pipe through which all traffic from peripheral networks must pass on its way to other peripheral networks.

**PE router**—provider edge router. The label edge router (LER) in the service provider network that connects to the customer edge (CE) router.

**POP**—point of presence. An access point to the Internet. A POP has a unique IP address. The ISP or online service provider (such as AOL) has one or more POPs on the Internet. ISP users dial into the POP to connect to the Internet. A POP can reside in rented space owned by the telecommunications carrier (such as Sprint) to which the ISP is connected. A POP usually includes routers, digital/analog call aggregators, servers, and frequently frame relay or ATM switches.

**RR**—route reflector. A router that advertises, or reflects, IBGP learned routes to other IBGP peers without requiring a full network mesh.

**VPN**—Virtual Private Network. A group of sites that, as a result of a set of administrative policies, can communicate with each other over a shared backbone.

**VPNv4 addresses**—When multiple VPNs use the same address space, the VPN addresses are made unique by adding a route distinguisher to the front of the address.

**VRF table**—VPN routing/forwarding table. A VRF table includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router. A VRF table consists of the following elements:

- An IP routing table
- A derived forwarding table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocols that determine what goes into the forwarding table

**Note**

---

Refer to the [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

---

---

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Copyright © 2005 Cisco Systems, Inc. All rights reserved.

