Configuring MPLS VPN Route Target Rewrite

- Finding Feature Information, on page 1
- Prerequisites for MPLS VPN Route Target Rewrite, on page 1
- Restrictions for MPLS VPN Route Target Rewrite, on page 1
- Information About MPLS VPN Route Target Rewrite, on page 2
- How to Configure MPLS VPN Route Target Rewrite, on page 3
- Configuration Examples for MPLS VPN Route Target Rewrite, on page 9

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for MPLS VPN Route Target Rewrite

- You should know how to configure Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).
- You need to identify the RT replacement policy and target device for the autonomous system (AS).

Restrictions for MPLS VPN Route Target Rewrite

Route Target Rewrite can only be implemented in a single AS topology.

ip unnumbered command is not supported in MPLS configuration.

Information About MPLS VPN Route Target Rewrite

Route Target Replacement Policy

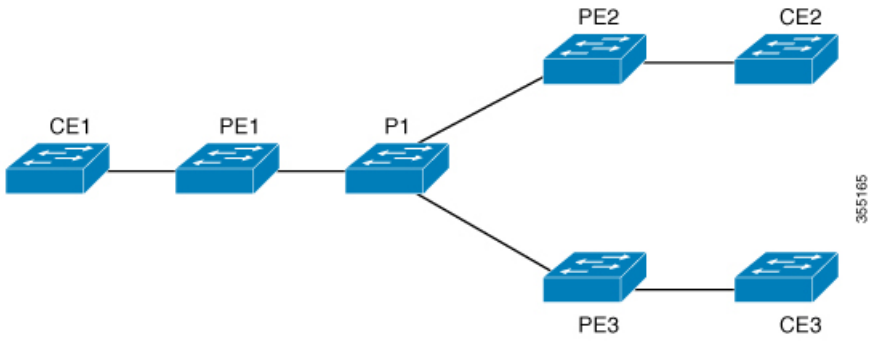
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound Border Gateway Protocol (BGP) updates. Route targets are carried as extended community attributes in BGP Virtual Private Network IP Version 4 (VPNv4) updates. Route target extended community attributes are used to identify a set of sites and VPN routing and forwarding (VRF) instances that can receive routes with a configured route target.

You can configure the MPLS VPN Route Target Rewrite feature on provider edge (PE) devices.

The figure below shows an example of route target replacement on PE devices in an Multiprotocol Label Switching (MPLS) VPN single autonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 65000:1 for VRF Customer A and to rewrite all inbound VPNv4 prefixes with RT 65000:1 to RT 65000:2.
- PE2 is configured to import and export RT 65000:2 for VRF Customer B and to rewrite all inbound VPNv4 prefixes with RT 65000:2 to RT 65000:1.

Figure 1: Route Target Replacement on Provide Edge(PE) devices in a single MPLS VPN Autonomous System Topology



Route Maps and Route Target Replacement

The MPLS VPN Route Target Rewrite feature extends the Border Gateway Protocol (BGP) inbound/outbound route map functionality to enable route target replacement. The **set extcomm-list delete** command entered in route-map configuration mode allows the deletion of a route target extended community attribute based on an extended community list.

How to Configure MPLS VPN Route Target Rewrite

Configuring a Route Target Replacement Policy

Perform this task to configure a route target (RT) replacement policy for your internetwork.

If you configure a provider edge (PE) device to rewrite RT *x* to RT *y* and the PE has a virtual routing and forwarding (VRF) instance that imports RT *x* , you need to configure the VRF to import RT *y* in addition to RT *x* .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match extcommunity** {*standard-list-number* | *expanded-list-number*}
6. **set extcomm-list** *extended-community-list-number* **delete**
7. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
8. **end**
9. **show route-map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*] **Example:** `Device(config)# ip extcommunity-list 1 permit rt 65000:2`	Creates an extended community access list and controls access to it. • The *standard-list-number* argument is an integer from 1 to 99 that identifies one or more permit or deny groups of extended communities. • The *expanded-list-number* argument is an integer from 100 to 500 that identifies one or more permit or deny groups of extended communities. Regular expressions can be configured with expanded lists but not standard lists.

	Command or Action	Purpose

		• The **permit** keyword permits access for a matching condition.
		• The **deny** keyword denies access for a matching condition.
		• The *regular-expression* argument specifies an input string pattern to match against. When you use an expanded extended community list to match route targets, include the pattern RT: in the regular expression.
		• The **rt** keyword specifies the route target extended community attribute. The **rt** keyword can be configured only with standard extended community lists and not expanded community lists.
		• The **soo** keyword specifies the site of origin (SOO) extended community attribute. The **soo** keyword can be configured only with standard extended community lists and not expanded community lists.
		• The *extended-community-value* argument specifies the route target or site of origin. The value can be one of the following combinations:
		  • autonomous-system-number:network-number
		  • ip-address:network-number
		The colon is used to separate the autonomous system number and network number or IP address and network number.

Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map rtrewrite permit 10`	Defines the conditions for redistributing routes from one routing protocol into another or enables policy routing and enables route-map configuration mode. • The *map-name* argument defines a meaningful name for the route map. The **redistribute** router configuration command uses this name to reference this route map. Multiple route maps can share the same map name. • If the match criteria are met for this route map, and the **permit** keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed. If the match criteria are not met, and the **permit** keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set.

	Command or Action	Purpose
		The **permit** keyword is the default. • If the match criteria are met for the route map and the **deny** keyword is specified, the route is not redistributed. In the case of policy routing, the packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used. • The *sequence-number* argument is a number that indicates the position a new route map will have in the list of route maps already configured with the same name. If given with the **no** form of this command, the position of the route map should be deleted.

Step 5	**match extcommunity** {*standard-list-number* | *expanded-list-number*}
	Example:
	`Device(config-route-map)# match extcommunity 1`
	Example:
	`Device(config-route-map)# match extcommunity 101`
		Matches the Border Gateway Protocol (BGP) extended community list attributes.
		• The *standard-list-number* argument is a number from 1 to 99 that identifies one or more permit or deny groups of extended community attributes.
		• The *expanded-list-number* argument is a number from 100 to 500 that identifies one or more permit or deny groups of extended community attributes.

Step 6	**set extcomm-list** *extended-community-list-number* **delete** **Example:** `Device(config-route-map)# set extcomm-list 1 delete`	Removes a route target from an extended community attribute of an inbound or outbound BGP Virtual Private Network Version 4 (VPNv4) update. • The *extended-community-list-number* argument specifies the extended community list number.

Step 7	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
	Example:
	`Device(config-route-map)# set extcommunity rt 65000:1 additive`
		Sets BGP extended community attributes.
		• The **rt** keyword specifies the route target extended community attribute.
		• The **soo** keyword specifies the site of origin extended community attribute.
		• The *extended-community-value* argument specifies the value to be set. The value can be one of the following combinations:
		  • autonomous-system-number : network-number
		  • ip-address : network-number
		The colon is used to separate the autonomous system number and network number or IP address and network number.

	Command or Action	Purpose
		• The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.
Step 8	**end** **Example:** `Device(config-route-map)# end`	(Optional) Returns to privileged EXEC mode.
Step 9	**show route-map** *map-name* **Example:** `Device# show route-map extmap`	(Optional) Verifies that the match and set entries are correct. • The *map-name* argument is the name of a specific route map.

Applying the Route Target Replacement Policy

Perform the following tasks to apply the route target replacement policy to your network:

Associating Route Maps with Specific BGP Neighbors

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a Border Gateway Protocol (BGP) routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 172.10.0.2 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard Virtual Private Network Version 4 (VPNv4) address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 172.16.0.2 activate`	Enables the exchange of information with a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 172.16.0.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The **both** keyword sends standard and extended community attributes. • The **extended** keyword sends an extended community attribute.

	Command or Action	Purpose
		• The **standard** keyword sends a standard community attribute.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 172.16.0.2 route-map extmap in`	Apply a route map to incoming or outgoing routes • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP or multiprotocol peer group. • The *map-name* argument specifies the name of a route map. • The **in** keyword applies route map to incoming routes. • The **out** keyword applies route map to outgoing routes.
Step 9	**end** **Example:** `Device(config-router-af)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the Route Target Replacement Policy

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 vrf** *vrf-name*
3. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show ip bgp vpnv4 vrf** *vrf-name*

Verifies that Virtual Private Network Version 4 (VPNv4) prefixes with a specified route target (RT) extended community attribute are replaced with the proper RT extended community attribute to verify that the provider edge (PE) devices receive the rewritten RT extended community attributes.

Verify route target replacement on PE1:

Example:

```
Device# show ip bgp vpnv4 vrf Customer_A 192.168.1.1/32 internal
BGP routing table entry for 65000:1:192.168.1.1/32, version 6901
Paths: (1 available, best #1, table Customer_A)
  Advertised to update-groups:
     5
  Refresh Epoch 1
  650002
    3.3.3.3 (metric 3) (via default) from 3.3.3.3 (55.5.4.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:65000:1
      mpls labels in/out nolabel/3025
      rx pathid: 0, tx pathid: 0x0
      net: 0xFFB0A72E38, path: 0xFFB0E6A370, pathext: 0xFFB0E5D970
      flags: net: 0x0, path: 0x7, pathext: 0x181
```

Step 3 **exit**

Returns to user EXEC mode:

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS VPN Route Target Rewrite

Examples: Configuring Route Target Replacement Policies

This example shows the route target (RT) replacement configuration of a Provider Edge (PE) device that exchanges Virtual Private Network Version 4 (VPNv4) prefixes with another Provider Edge (PE) device. The route map extmap is configured to replace RTs on inbound updates. Any incoming update with RT 65000:2 is replaced with RT 65000:1.

```
!
ip extcommunity-list 1 permit rt 65000:2
!
route-map rtrewrite permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 65000:1 additive
!
```

This example shows the use of the route-map configuration **continue** command when you need to apply more than one replacement rule on an update. In this example, an incoming update with RT 7777:222222222 is replaced with RT 65000:2. Without the **continue 20** command, route-map evaluation would stop when a match on sequence 10 is made. With the **continue 20** command, route-map evaluation continues into sequence 20 even if a match occurs in sequence 10.

```
!
ip extcommunity-list 2 permit rt 7777:222222222
ip extcommunity-list 3 permit rt 2:2
ip extcommunity-list 4 permit rt 20000:111
```

```
!
route-map extmap1 permit 10
match extcommunity 2
continue 20
set extcomm-list 2 delete
set extcommunity rt 65000:2 additive
!
route-map extmap1 permit 20
match extcommunity 3
continue 30
set extcomm-list 3 delete
!
route-map extmap1 permit 30
match extcommunity 4
set extcomm-list 4 delete
!
```

Note The route-map configuration **continue** command is not supported on outbound route maps.

Examples: Applying Route Target Replacement Policies

Examples: Associating Route Maps with Specific BGP Neighbor

This example shows the association of route map extmap with a Border Gateway Protocol (BGP) neighbor. The BGP inbound route map is configured to replace route targets (RTs) on incoming updates.

```
router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite in
```

This example shows the association of the same route map with the outbound BGP neighbor. The route map is configured to replace RTs on outgoing updates.

```
router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite out
```