MPLS VPN—Route Target Rewrite

First Published: August 26, 2003
Last Updated: May 31, 2007

The MPLS VPN—Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) routers can also perform route target replacement.

The main advantage of the MPLS VPN - Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system.

History for the MPLS VPN - Route Target Rewrite Feature

Release	Modification
12.0(26)S	This feature was introduced for the Cisco 7200, 7500 and 12000 series routers.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S to support the Cisco 7500 series router.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.0(33)S	The Route Target Rewrite at autonomous system Boundaries feature is now supported on an IP core using L2TPv3 on Cisco 12000 series routers.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

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

Contents

- Prerequisites for MPLS VPN - Route Target Rewrite, page 2
- Restrictions for MPLS VPN - Route Target Rewrite, page 2
- Information About MPLS VPN - Route Target Rewrite, page 2
- How to Configure MPLS VPN - Route Target Rewrite, page 4
- Configuration Examples for MPLS VPN - Route Target Rewrite, page 16
- Additional References, page 21
- Command Reference, page 22
- Glossary, page 25

Prerequisites for MPLS VPN - Route Target Rewrite

The MPLS VPN - Route Target Rewrite feature requires the following:

- You should know how to configure Multiprotocol Virtual Private Networks (MPLS VPNs).
- You need to configure your network to support interautonomous systems (Inter-autonomous system) with different route target (RT) values in each autonomous system.
- You need to identify the RT replacement policy and target router for each autonomous system.

Restrictions for MPLS VPN - Route Target Rewrite

You can apply multiple replacement rules using the route-map continue clause. The MPLS VPN - Route Target Rewrite feature does not support the continue clause on outbound route maps.

Information About MPLS VPN - Route Target Rewrite

To configure the MPLS VPN - Route Target Rewrite feature, you need to understand the following concepts:

- Route Target Replacement Policy, page 2
- Route Maps and Route Target Replacement, page 4

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 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.

In general, ASBRs perform route target replacement at autonomous system borders when the ASBRs exchange VPNv4 prefixes. You can also configure the MPLS VPN - Route Target Rewrite feature on PE routers and RR routers.

Figure 1 shows an example of route target replacement on ASBRs in an MPLS VPN Inter-autonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- ASBR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 to RT 100:1.
- ASBR2 is configured to rewrite all inbound VPNv4 prefixes with RT 100:1 to RT 200:1.

Figure 1 Route Target Replacement on ASBRs in an MPLS VPN Inter-AS Topology

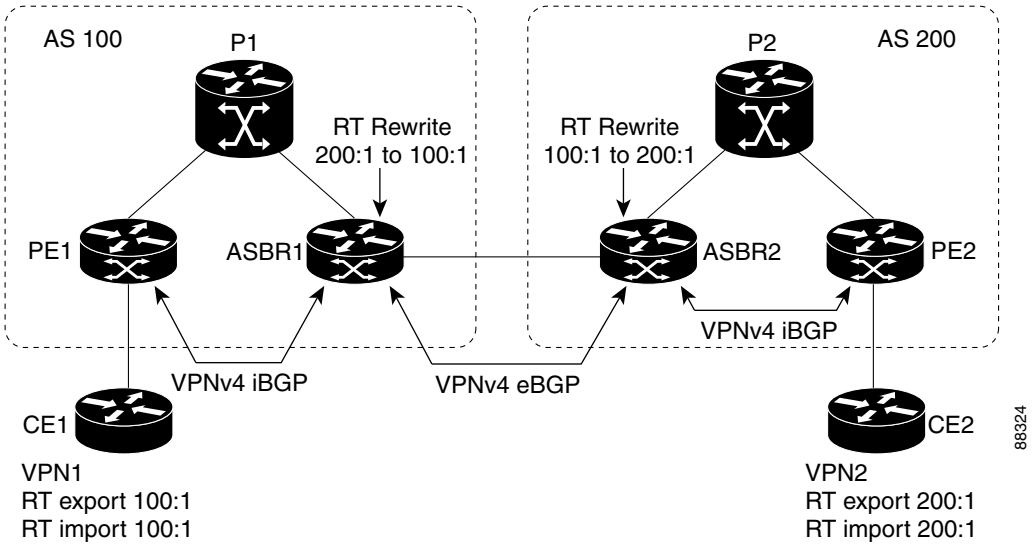
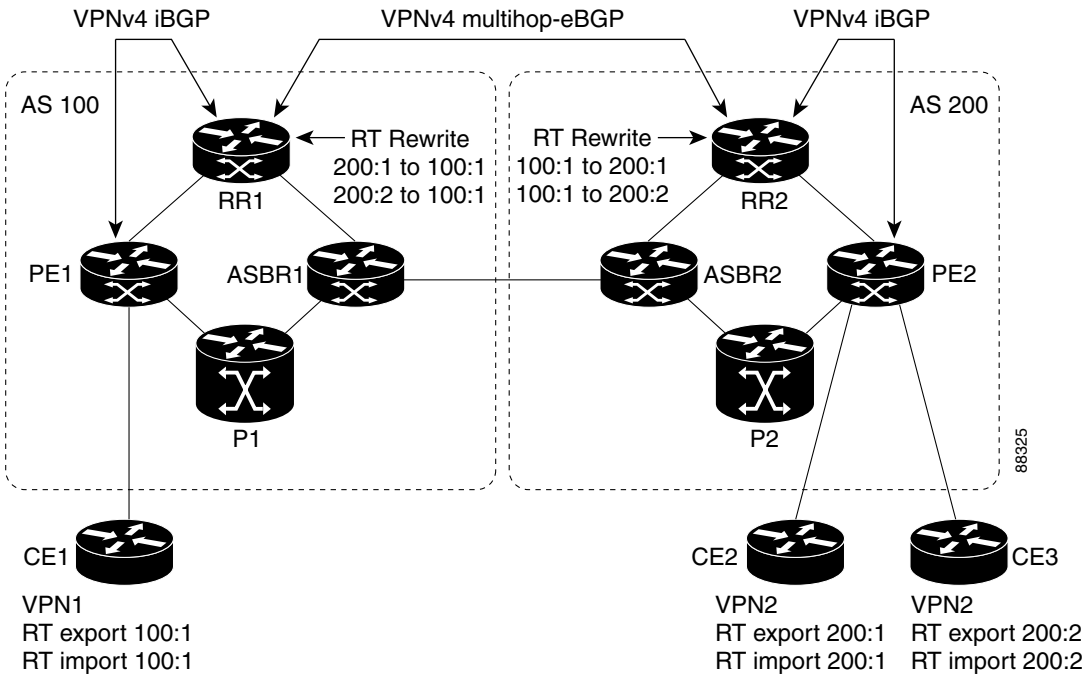


Figure 2 shows an example of route target replacement on route reflectors in an MPLS VPN Inter-autonomous system topology. This example includes the following configurations:

- EBGP is configured on the route reflectors.
- EBGP and IBGP IPv4 label exchange is configured between all BGP routers.
- Peer groups are configured on the routers reflectors.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- PE2 is configured to import and export RT 200:2 for VRF VPN3.
- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- RR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 or RT 200:2 to RT 100:1.
- RR2 is configured to rewrite all inbound prefixes with RT 100:1 to RT 200:1 and RT 200:2.

Figure 2 Route Target Rewrite on Route Reflectors in an MPLS VPN Inter-autonomous system Topology



Route Maps and Route Target Replacement

The MPLS VPN - Route Target Rewrite feature extends the 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

This section contains the following procedures to configure MPLS VPN - Route Target Rewrite:

- Configuring a Route Target Replacement Policy, page 4 (required)
- Applying the Route Target Replacement Policy, page 9 (required)
- Verifying the Route Target Replacement Policy, page 13 (optional)
- Troubleshooting Your Route Target Replacement Policy, page 14 (optional)

Configuring a Route Target Replacement Policy

Perform this task to configure an RT replacement policy for your internetwork.

If you configure a PE to rewrite RT x to RT y and the PE has a VRF 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-tag* [**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**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*] **Example:** `Router(config)# ip extcommunity-list 1 permit rt 100:3`	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. • 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.

	Command or Action	Purpose
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map extmap 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-tag* 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 may share the same map tag 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. 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:
	`Router(config-route-map)# match extcommunity 1`
	
	Example:
	`Router(config-route-map)# match extcommunity 101`
		Matches 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:** `Router(config-route-map)# set extcomm-list 1 delete`	Removes a route target from an extended community attribute of an inbound or outbound BGP VPNv4 update. • The *extended-community-list-number* argument specifies the extended community list number.

	Command or Action	Purpose
Step 7	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*} **Example:** `Router(config-route-map)# set extcommunity rt 100:4 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. • The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.
Step 8	**end** **Example:** `Router(config-route-map)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Use the **show route-map map-name** command to verify that the match and set entries are correct.

Applying the Route Target Replacement Policy

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

- Associating Route Maps with Specific BGP Neighbors, page 9
- Refreshing BGP Session to Apply Route Target Replacement Policy, page 11

Associating Route Maps with Specific BGP Neighbors

Perform this task to associate 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

	Command or Action	Purpose
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(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:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**] **Example:** `Router(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. • The **standard** keyword sends a standard community attribute.

	Command or Action	Purpose
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(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:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Refreshing BGP Session to Apply Route Target Replacement Policy

Perform this task to refresh the BGP session to apply the RT replacement policy.

After you have defined two routers to be BGP neighbors, the routers form a BGP connection and exchange routing information. If you subsequently change a routing policy, you must reset BGP connections for the configuration change to take effect. After configuring the RT replacement policy and applying it to the target routers in your system, you must refresh the BGP session to put the policy into operation.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | **in** | **out**}]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | **in** | **out**}] **Example:** `Router# clear ip bgp vpnv4 unicast 172.16.0.2 in`	Resets a BGP connection using BGP soft reconfiguration. • The * keyword resets all current BGP sessions. • The *neighbor-address* argument resets only the identified BGP neighbor. • The *peer-group-name* argument resets the specified BGP peer group. • The **ipv4** keyword resets the specified IPv4 address family neighbor or peer group. The **multicast** or **unicast** keyword must be specified. • The **vpnv4** keyword resets the specified VPNv4 address family neighbor or peer group. The **unicast** keyword must be specified. • The **soft** keyword indicates a soft reset. Does not reset the session. The **in** or **out** keywords do not follow the **soft** keyword when a connection is cleared under the VPNv4 or IPv4 address family because the **soft** keyword specifies both. • The **in** and **out** keywords trigger inbound or outbound soft reconfiguration, respectively. If the **in** or **out** keyword is not specified, both inbound and outbound soft reset are triggered.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Troubleshooting Tips

To determine whether a BGP router supports the route refresh capability, use the **show ip bgp neighbors** command. If a router supports the route refresh capability, the following message is displayed:

```
Received route refresh capability from peer.
```

You can issue the **debug ip bgp updates** command on the router where you entered the **clear ip bgp** command to verify that the updates are occurring.

Note Issuing the **debug ip bgp updates** command could impair performance if the router sends or receives a large number of BGP updates.

Verifying the Route Target Replacement Policy

Perform this task to verify the operation of your RT replacement policy.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix*/*length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** `Router# show ip bgp vpnv4 all 172.16.0.2`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4** command with the **all** keyword to verify that all VPNv4 prefixes with a specified RT extended community attribute are replaced with the proper RT extended community attribute at the ASBRs or route reflectors and to verify that the PE routers receive the rewritten RT extended community attributes from the ASBRs or route reflectors.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Troubleshooting Your Route Target Replacement Policy

Perform this task to troubleshoot your RT replacement policy.

SUMMARY STEPS

1. **enable**
2. **debug ip bgp** [*A.B.C.D.* | **dampening** | **events** | **in** | **keepalives** | **out** | **updates** | **vpnv4** | **mpls**]
3. **clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | **in** | **out**}]
4. **debug ip bgp** [*A.B.C.D.* | **dampening** | **events** | **in** | **keepalives** | **out** | **updates** | **vpnv4** | **mpls**]
5. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix/length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
6. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip bgp** [*A.B.C.D.*	**dampening**	**events**	**in**	**keepalives**	**out**	**updates**	**vpnv4**	**mpls**] **Example:** `Router# debug ip bgp updates`	(Optional) Displays information related to processing of the BGP protocol. • Use the **debug ip bgp updates** command to verify that BGP updates are occurring. **Note** Issuing the **debug ip bgp updates** command could impair performance if the router sends or receives a large number of BGP updates.

	Command or Action	Purpose
Step 3	**clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | **in** | **out**}] **Example:** `Router# clear ip bgp *`	(Optional) Resets a BGP connection using BGP soft reconfiguration. • The ***** keyword resets all current BGP sessions. • The *neighbor-address* argument resets only the identified BGP neighbor. • The *peer-group-name* argument resets the specified BGP peer group. • The **ipv4** keyword resets the specified IPv4 address family neighbor or peer group. The **multicast** or **unicast** keyword must be specified. • The **vpnv4** keyword resets the specified VPNv4 address family neighbor or peer group. The **unicast** keyword must be specified. • The **soft** keyword indicates a soft reset. Does not reset the session. The **in** or **out** keywords do not follow the **soft** keyword when a connection is cleared under the VPNv4 or IPv4 address family because the **soft** keyword specifies both. • The **in** and **out** keywords trigger inbound or outbound soft reconfiguration, respectively. If the **in** or **out** keyword is not specified, both inbound and outbound soft reset are triggered.
Step 4	**debug ip bgp** [*A.B.C.D.* | **dampening** | **events** | **in** | **keepalives** | **out** | **updates** | **vpnv4** | **mpls**] **Example:** `Router# debug ip bgp updates`	(Optional) Displays information related to processing of the BGP protocol. • Use the **debug ip bgp updates** command to verify that BGP updates are occurring after the **clear ip bgp** command. **Note** Issuing the **debug ip bgp updates** command could impair performance if the router sends or receives a large number of BGP updates.
Step 5	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix*/*length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [*community*] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** `Router# show ip bgp vpnv4 all 172.10.0.2`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4** command with the **all** keyword to verify that RT extended community attribute are replaced correctly.
Step 6	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN - Route Target Rewrite

This section contains the following configuration examples for the MPLS VPN - Route Target Rewrite feature:

- Configuring Route Target Replacement Policies: Examples, page 16
- Applying Route Target Replacement Policies: Examples, page 17
- Verifying the Route Target Replacement Policy Example, page 18
- Troubleshooting the Route Target Replacement Policy Example, page 19

Configuring Route Target Replacement Policies: Examples

This example shows the RT replacement configuration of an ASBR (ASBR1) that exchanges VPNv4 prefixes with another ASBR (ASBR2). The route map extmap is configured to replace RTs on inbound updates. Any incoming update with RT 100:3 is replaced with RT 200:3. Any other prefixes with an RT whose autonomous system number is 100 is rewritten to RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 101 permit RT:100:*
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
!
route-map regexp permit 10
match extcommunity 101
set extcomm-list 101 delete
set extcommunity rt 200:4 additive
!
route-map regexp permit 20
```

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 100:3 is replaced with RT 200:3. 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. If the incoming update has an RT 100:4, the router replaces it with RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 2 permit rt 100:4
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
continue 20
!
route-map extmap permit 20
match extcommunity 2
set extcomm-list 2 delete
set extcommunity rt 200:4 additive
!
route-map extmap permit 30
```

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

Applying Route Target Replacement Policies: Examples

This section contains the following examples:

- Associating Route Maps with Specific BGP Neighbor Example, page 17
- Refreshing the BGP Session to Apply the Route Target Replacement Policy Example, page 17

Associating Route Maps with Specific BGP Neighbor Example

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

```
router bgp 100
.
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap 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 100
.
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap out
```

Refreshing the BGP Session to Apply the Route Target Replacement Policy Example

The following example shows the **clear ip bgp** command used to initiate a dynamic reconfiguration in the BGP peer 172.16.0.2. This command requires that the peer supports the route refresh capability.

```
Router# clear ip bgp 172.16.0.2 vpnv4 unicast in
```

Verifying the Route Target Replacement Policy Example

The following examples verify route target replacement on ABSR1 and ABSR2.

Verify route target replacement on ABSR1:

```
Router# show ip bgp vpnv4 all 172.16.17.17

BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  300
    172.16.11.11 (metric 589) from 172.16.11.11 (172.16.11.11)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:200:1
```

Verify route target replacement on ABSR2:

```
Router# show ip bgp vpnv4 all 172.16.17.17

BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

The following examples verify route target replacement on PE1 and PE2.

Verify route target on PE1:

```
Router# show ip bgp vpnv4 all 172.16.17.17

BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     1
  300
    192.168.2.1 (via vpn1) from 192.168.2.1 (172.16.19.19)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:200:1
```

Verify route target on PE2:

```
Router# show ip bgp vpnv4 all 172.16.17.17

BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     3
  100 300
    192.168.1.1 (metric 20) from 172.16.16.16 (172.16.16.16)
      Origin incomplete, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Troubleshooting the Route Target Replacement Policy Example

Note Issuing the **debug ip bgp updates** command could impair performance if the router sends or receives a large number of BGP updates.

This example shows the BGP update information on ASBR1:

```
Router# debug ip bgp updates 172.16.16.16

BGP(2): no valid path for 100:1:172.16.20.20/32

BGP(2): no valid path for 100:1:10.0.0.0/8

%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Down User reset

BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB

BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB

BGP(2): 172.16.11.11 computing updates, afi 2, neighbor version 13,
 table version 15, starting at 0.0.0.0

BGP(2): 172.16.11.11 send unreachable 100:1:172.16.20.20/32

BGP(2): 172.16.11.11 send UPDATE 100:1:172.16.20.20/32 -- unreachable

BGP(2): 172.16.11.11 send UPDATE 100:1:192.168.3.0/8 -- unreachable

BGP(2): 1 updates (average = 58, maximum = 58)

BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11

BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15

BGP: Import walker start version 13, end version 15

BGP: ... start import cfg version = 30

%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Up

BGP(2): 172.16.16.16 computing updates, afi 2, neighbor version 0,
table version 15, starting at 0.0.0.0

BGP(2): 172.16.16.16 send UPDATE (format) 100:1:172.16.0.0/16,
next 172.16.11.11, metric 0, path 300, extended community RT:2:2
RT:7777:222222222 RT:20000:111 RT:65535:999999999

BGP(2): 172.16.16.16 send UPDATE (prepend, chgflags: 0x0)
100:1:172.16.19.19/32, next 172.16.11.11, metric 0, path 300,
extended community RT:2:2 RT:7777:222222222 RT:20000:111
RT:65535:999999999

BGP(2): 172.16.16.16 send UPDATE (format) 100:1:192.168.2.0/8,
next 172.16.11.11, metric 0, path , extended community
RT:2:2 RT:7777:222222222 RT:20000:111 RT:65535:999999999

BGP(2): 2 updates (average = 111, maximum = 121)

BGP(2): 172.16.16.16 updates replicated for neighbors: 172.16.16.16
```

```
BGP(2): 172.16.16.16 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15

BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200, extended community RT:100:1

BGP(2): 172.16.16.16 rcvd 100:1:192.168.3.0/8

BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200 400, extended community RT:100:1

BGP(2): 172.16.16.16 rcvd 100:1:172.16.0.0/16

BGP(2): 172.16.16.16 rcvd 100:1:172.16.20.20/32

BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB

BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB

BGP: Import walker start version 15, end version 17

BGP: ... start import cfg version = 30

BGP(2): 172.16.11.11 computing updates, afi 2,
neighbor version 15, table version 17,
starting at 0.0.0.0

BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:172.16.20.20/32,
next 172.16.15.15

BGP(2): 172.16.11.11 send UPDATE (format) 100:1:172.16.20.20/32,
next 172.16.15.15,metric 0, path 200 400, extended community
RT:1:1 RT:10000:111 RT:33333:888888888
RT:65535:999999999

BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:10.0.0.0/8,
next 172.16.15.15

BGP(2): 172.16.11.11 send UPDATE (format) 100:1:192.168.3.0/8,
next 172.16.15.15, metric 0, path 200, extended community
RT:1:1 RT:10000:111 RT:33333:888888888 RT:65535:999999999

BGP(2): 2 updates (average = 118, maximum = 121)

BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11

BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 17, start version 17, throttled to 17
```

This example shows VPN address information from the BGP table and verifies that RT extended community attributes are replaced correctly:

```
Router# show ip bgp vpnv4 all 172.16.17.17

BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

Additional References

The following sections provide references related to the MPLS VPN - Route Target Rewrite feature.

Related Documents

Related Topic	Document Title
MPLS VPN interautonomous systems configuration tasks	*MPLS VPN—Interautonomous System Support*
VPN configuration tasks	*MPLS Virtual Private Networks (VPNs)*
BGP configuration tasks	*Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4
MPLS configuration tasks	*Cisco IOS Multiprotocol Label Switching Configuration Guide*, Release 12.4
Commands to configure and monitor BGP	• *Cisco IOS IP Routing Protocols Command Reference*, Release 12.4T • *Cisco IOS IP Routing Protocols Command Reference*, Release 12.2SB • *Cisco IOS IP Routing Protocols Command Reference*, Release 12.2 SR

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: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Command Reference

This section documents only commands that are new or modified.

- **set extcomm-list delete**

set extcomm-list delete

To allow the deletion of extended community attributes based on an extended community list, use the **set extcomm-list delete** command in route-map configuration mode. To negate a previous **set extcomm-list detect** command, use the **no** form of this command.

set extcomm-list *extended-community-list-number* **delete**

no set extcomm-list *extended-community-list-number* **delete**

Syntax Description

extended-community-list-number	An extended community list number.

Command Modes

Route-map configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command removes extended community attributes of an inbound or outbound Border Gateway Protocol (BGP) update using a route map to filter and determine the extended community attribute to be deleted and replaced. Depending upon whether the route map is applied to the inbound or outbound update for a neighbor, each extended community that passes the route map permit clause and matches the given extended community list will be removed and replaced from the extended community attribute being received from or sent to the BGP neighbor.

Examples

The following example shows how to replace a route target 100:3 on an incoming update with a route target of 100:4 using an inbound route map extmap:

```
.
.
.
Router(config-af)# neighbor 10.10.10.10 route-map extmap in
.
.
.
Router(config)# ip extcommunity-list 1 permit rt 100:3
Router(config)# route-map extmap permit 10
Router(config-route-map)# match extcommunity 1
Router(config-route-map)# set extcomm-list 1 delete
Router(config-route-map)# set extcommunity rt 100:4 additive
```

The following example shows how to configure more than one replacement rule using the route-map configuration **continue** command. Prefixes with RT 100:2 are rewritten to RT 200:3 and prefixes with RT 100:4 are rewritten to RT 200:4. With the **continue** command, route-map evaluation proceeds even if a match is found in a previous sequence.

```
Router(config)# ip extcommunity-list 1 permit rt 100:3
Router(config)# ip extcommunity-list 2 permit rt 100:4
Router(config)# route-map extmap permit 10
Router(config-route-map)# match extcommunity 1
Router(config-route-map)# set extcomm-list 1 delete
Router(config-route-map)# set extcommunity rt 200:3 additive
Router(config-route-map)# continue 20
Router(config)# route-map extmap permit 20
Router(config-route-map)# match extcommunity 2
Router(config-route-map)# set extcomm-list 2 delete
Router(config-route-map)# set extcommunity rt 200:4 additive
Router(config-route-map)# exit
Router(config)# route-map extmap permit 30
```

Related Commands

Command	Description
ip community-list	Creates an extended community access list and controls access to it.
match extcommunity	Matches BGP extended community list attributes.
route-map (IP)	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.
set extcommunity	Sets BGP extended community attributes.

Glossary

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

ASBR—autonomous system border router. A router that connects and exchanges information between two or more autonomous systems.

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.

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. When a pair of routers in different autonomous systems are more than one IP hop away from each other, an EBGP session between those two routers is called multihop EBGP.

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 Internal Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

LDP—Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

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

LSR—label switch router. The role of an LSR is to forward packets in an MPLS network by looking only at the fixed-length label.

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

NLRI—Network Layer Reachability Information. BGP sends routing update messages containing NLRI, 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.

RD—route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

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

RT—route target. Extended community attribute used to identify the VRF routing table into which a prefix is to be imported.

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 prefix—IPv4 prefix preceded by an 8-byte route distinguisher. The VPN addresses are made unique by adding a route distinguisher to the front of the address.

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

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