Configuring BGP-VPN Distinguisher Attribute

- Information About BGP-VPN Distinguisher Attribute, on page 1
- How to Configure BGP-VPN Distinguisher Attribute, on page 3
- Configuration Example for Translating RT to VPN Distinguisher to RT, on page 8
- Feature History for BGP-VPN Distinguisher Attribute, on page 9

Information About BGP-VPN Distinguisher Attribute

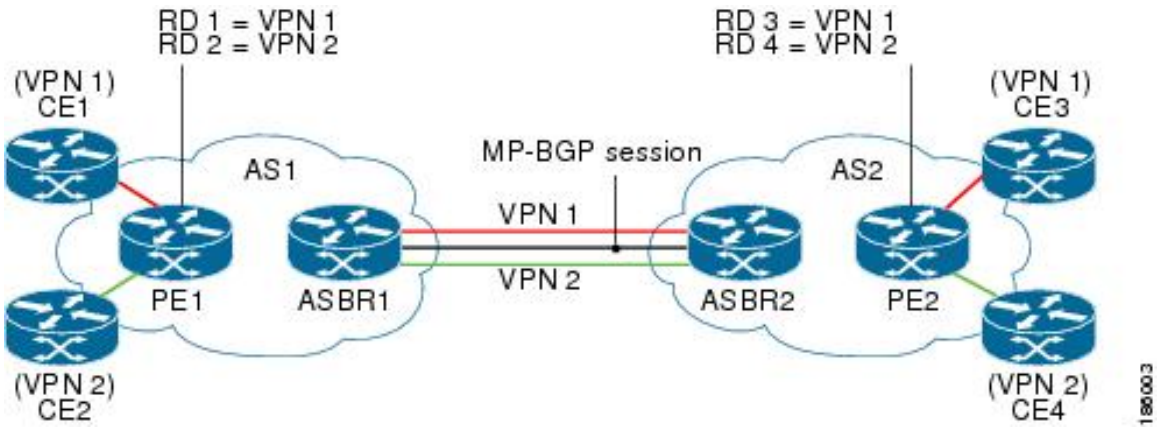
The following sections provide information about BGP-VPN distinguisher attribute.

Role and Benefit of the VPN Distinguisher Attribute

Route-target (RT) extended community attributes identify the VPN membership of routes. The RT attributes are placed onto a route at the exporting (egress) provider edge router (PE) and are transported across the iBGP cloud and across autonomous systems. Any Virtual Routing and Forwarding (VRF) instances at the remote PE that want to import such routes must have the corresponding RTs set as import RTs for that VRF.

The figure below illustrates two autonomous systems, each containing customer edge routers (CEs) that belong to different VPNs. Each PE tracks which route distinguisher (RD) corresponds to which VPN, thus controlling the traffic that belongs to each VPN.

Figure 1: Scenario in Which ASBRs Translate RTs Between Autonomous Systems



In an Inter-AS Option B scenario like the one in the figure above, these routes are carried across an AS boundary from Autonomous System Border Router 1 (ASBR1) to ASBR2 over an MP-eBGP session, with the routes' respective RTs as extended community attributes being received by ASBR2.

ASBR2 must maintain complex RT mapping schemes to translate RTs originated by AS1 to RTs recognized by AS2, so that the RTs can be imported by their respective VPN membership CE connections on PE2 for CE3 and CE4.

Some network administrators prefer to hide the RTs they source in AS1 from devices in AS2. In order to do that, the administrator must differentiate routes belonging to each VPN with a certain attribute so that the RTs can be removed on the outbound side of ASBR1 before sending routes to ASBR2, and ASBR2 can then map that attribute to recognizable RTs in AS2. The VPN Distinguisher (VD) extended community attribute serves that purpose.

The benefit of the BGP—VPN Distinguisher Attribute feature is that source RTs can be kept private from devices in destination autonomous systems.

How the VPN Distinguisher Attribute Works

The network administrator configures the egress ASBR to perform translation of RTs to a VPN distinguisher extended community attribute, and configures the ingress ASBR to perform translation of the VPN distinguisher to RTs. More specifically, the translation is achieved as follows:

On the Egress ASBR

- An outbound route map specifies a **match excommunity** clause that determines which VPN routes are subject to mapping, based on the route's RT values.
- A **set extcommunity vpn-distinguisher** command sets the VPN distinguisher that replaces the RTs.
- The **set extcomm-list delete** command that references the same set of RTs is configured to remove the RTs, and then the route is sent to the neighboring ingress ASBR.

On the Ingress ARBR

- An inbound route map specifies a **match excommunity vpn-distinguisher** command that determines which VPN routes are subject to mapping, based on the route's VPN distinguisher.
- The **set extcommunity rt** command specifies the RTs that replace the VPN distinguisher.
- For routes that match the clause, the VPN distinguisher is replaced with the configured RTs.

Additional Behaviors Related to the VPN Distinguisher

On the egress ASBR, if a VPN route matches a route map clause that does not have the **set extcommunity vpn-distinguisher** command that is configured, the RTs that the VPN route is tagged with are retained.

The VPN distinguisher is transitive across the AS boundary, but is not carried within the iBGP cloud. That is, the ingress ASBR can receive the VPN distinguisher from an eBGP peer, but the VPN distinguisher is discarded on the inbound side after it is mapped to the corresponding RTs.

On the ingress ASBR, if a VPN route carrying the VPN distinguisher matches a route map clause that does not have a **set extcommunity rt** command that is configured in the inbound route map, the system does not discard the attribute, nor does it propagate the attribute within the iBGP cloud. The VPN distinguisher for the route is retained so that the network administrator can configure the correct inbound policy to translate the VPN distinguisher to the RTs that the VPN route should carry. If the route is sent to eBGP peers, the VPN

distinguisher is carried as is. The network administrator could configure a route-map entry to remove the VPN distinguisher from routes that are sent to eBGP peers.

Configuring a **set extcommunity vpn-distinguisher** command in an outbound route map or a **match excommunity** command in an inbound route map results in an outbound or inbound route refresh request, respectively, in order to update the routes being sent or received.

BGP-VPN Distinguisher Attribute

The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source route targets (RTs) private from an Autonomous System Border Router (ASBR) in a destination autonomous system. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.

How to Configure BGP-VPN Distinguisher Attribute

The following sections provide configuration information about BGP-VPN distinguisher attribute.

Replacing an RT with a VPN Distinguisher Attribute

Perform this task on an egress ASBR to replace a route target (RT) with a VPN distinguisher extended community attribute. Remember to replace the VPN distinguisher with a route target on the ingress ASBR; that task is described in the "Replacing a VPN Distinguisher Attribute with an RT" section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value* **Example:** `Device(config)# ip extcommunity-list 4 permit rt 101:100`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT are in the extended community list. This example permits routes having RT 101:100 into the extended community list 4.
Step 4	**exit** **Example:**	Exits the configuration mode and enters the next higher configuration mode.

	Command or Action	Purpose
	`Device(config-extcomm-list)# exit`	
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-map1 permit 10`	Configures a route map that permits or denies the routes that are allowed by the subsequent **match** command. This example permits the routes that are allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 4`	Matches on the specified community list. For this example, routes that match the extended community list 4 (which was configured in Step 3) are subject to the subsequent **set** commands.
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 4 delete`	Deletes the RT from routes that are in the specified extended community list. For this example, RTs are deleted from routes that are in extended community list 4.
Step 8	**set extcommunity vpn-distinguisher** *id* **Example:** `Device(config-route-map)# set extcommunity vpn-distinguisher 111:100`	For the routes that are permitted by the route map, sets the specified VPN distinguisher. For this example, routes that match extended community 4 have their VPN distinguisher set to 111:100.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-map1 permit 20`	(Optional) Configures a route map entry that permits routes. This example configures a route map entry that permits other routes not subject to the RT-to-VPN distinguisher mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 2000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.101.1 remote-as 2000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.101.1 activate`	Activates the specified neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Device(config-router-af)# neighbor 192.168.101.1 route-map vpn-id-map1 out`	Applies the specified outgoing route map to the specified neighbor.
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Replacing a VPN Distinguisher Attribute with an RT

Perform this task on an ingress ASBR to replace a VPN distinguisher extended community attribute with a route target (RT) attribute. This task assumes you already configured the egress ASBR to replace the RT with a VPN distinguisher; that task is described in the “Replacing an RT with a VPN Distinguisher Attribute” section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **vpn-distinguisher** *id* **Example:** `Device(config)# ip extcommunity-list 51 permit vpn-distinguisher 111:100`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified VPN distinguisher are in the extended community list. This example permits routes having VPN distinguisher 111:110 into the extended community list 51.
Step 4	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits the configuration mode and enters the next higher configuration mode.
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-rewrite-map1 permit 10`	Configures a route map that permits or denies the routes that are allowed by the subsequent **match** command. This example permits the routes that are allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 51`	Matches on the specified community list. For this example, routes that match the extended community list 51 (which was configured in Step 3) are subject to the subsequent **set** commands.
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 51 delete`	Deletes the VPN distinguisher from routes that are in the specified extended community list. For this example, VPN distinguishers are deleted from routes that are in extended community list 51.

	Command or Action	Purpose
Step 8	**set extcommunity rt** *value* **additive** **Example:** `Device(config-route-map)# set extcommunity rt 101:1 additive`	Sets the routes that are permitted by the route map with the specified RT. For this example, routes that match extended community 51 have their RT set to 101:1. The **additive** keyword causes the RT to be added to the RT list without replacing any RTs.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-rewrite-map1 permit 20`	(Optional) Configures a route map entry that permits routes. This example configures a route map entry that permits other routes not subject to the VPN distinguisher-to-RT mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 3000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.0.81 remote-as 3000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router-af)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.0.81 activate`	Activates the specified neighbor.

	Command or Action	Purpose
Step 16	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Device(config-router-af)# neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in`	Applies the specified outgoing route map to the specified neighbor.
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Example

Configuration Example for Translating RT to VPN Distinguisher to RT

The following example shows the egress ASBR configuration to replace a route target (RT) with a VPN distinguisher, and shows the ingress ASBR configuration to replace the VPN distinguisher with a route target.

On the egress ASBR, IP extended community list 1 is configured to filter VPN routes by permitting only routes with RT 101:100. A route map named vpn-id-map1 says that any route that matches on routes that are allowed by IP extended community list 1 are subject to two **set** commands. The first **set** command deletes the RT from the route. The second **set** command sets the VPN distinguisher attribute to 111:100.

The **route-map vpn-id-map1 permit 20** command allows other routes, which are not part of the RT-to-VPN distinguisher mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause these routes to be discarded.

Finally, in autonomous system 2000, for the VPNv4 address family, the route map vpn-id-map1 is applied to routes going out to the neighbor at 192.168.101.1.

Egress ASBR

```
ip extcommunity-list 1 permit rt 101:100
!
route-map vpn-id-map1 permit 10
 match extcommunity 1
 set extcomm-list 1 delete
 set extcommunity vpn-distinguisher 111:100
!
route-map vpn-id-map1 permit 20
!
router bgp 2000
```

```
 neighbor 192.168.101.1 remote-as 2000
 address-family vpnv4
  neighbor 192.168.101.1 activate
  neighbor 192.168.101.1 route-map vpn-id-map1 out
  exit-address-family
!
```

On the ingress ASBR, IP extended community list 51 allows routes with a VPN distinguisher of 111:100. A route map named vpn-id-rewrite-map1 says that any route that matches on routes that are allowed by IP extended community list 51 are subject to two **set** commands. The first **set** command deletes the VPN distinguisher from the route. The second **set** command sets the RT to 101:1, and that RT is added to the RT list without replacing any RTs.

The **route-map vpn-id-rewrite-map1 permit 20** command allows other routes, which are not part of the VPN distinguisher-to-RT mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause those routes to be discarded.

Finally, in autonomous system 3000, for the VPNv4 address family, the route map named vpn-id-rewrite-map1 is applied to incoming routes destined for the neighbor at 192.168.0.81.

Ingress ASBR

```
ip extcommunity-list 51 permit vpn-distinguisher 111:100
!
route-map vpn-id-rewrite-map1 permit 10
 match extcommunity 51
 set extcomm-list 51 delete
 set extcommunity rt 101:1 additive
!
route-map vpn-id-rewrite-map1 permit 20
!
router bgp 3000
 neighbor 192.168.0.81 remote-as 3000
 address-family vpnv4
  neighbor 192.168.0.81 activate
  neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in
  exit-address-family
!
```

Feature History for BGP-VPN Distinguisher Attribute

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	BGP-VPN Distinguisher Attribute	The BGP-VPN Distinguisher Attribute feature allows a network administrator to keep source route targets private from an ASBR in a destination autonomous system. Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Fuji 16.8.1a	BGP-VPN Distinguisher Attribute	Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Cupertino 17.7.1	BGP-VPN Distinguisher Attribute	Support for this feature was introduced on the C9500X-28C8D model of the Cisco Catalyst 9500 Series Switches.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/