Configuring VRF aware PBR

- Restrictions for VRF aware PBR, on page 1
- Information about VRf aware PBR, on page 1
- How to Configure VRF aware PBR, on page 3
- Configuration Examples for VRF aware PBR, on page 22
- Feature History for VRF aware PBR, on page 29

Restrictions for VRF aware PBR

- The route map commands **set global** and **set vrf** cannot be configured together on the same route-map.
- The same PBR cannot be applied to multiple unique VRF interfaces. The exception is when the PBR policy contains a **set global** or **set vrf** as the set command.
- Different route map command options (**set ip vrf**,**set ip default vrf**,**set vrf**) cannot be configured on the same route-map under the same sequence or a different sequence. Multiple unique route map command options (such as **set vrf**) can be configured using different sequence number in route-map.

Information about VRf aware PBR

Overview

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF.

Starting with Cisco IOS XE 16.12.1 release, PBR can be configured on VRF lite interfaces.

MPLS cannot be configured on the same VRF lite interface that has PBR configured on it.

VRF aware PBR can be of the following types:

- **Inherit VRF**: For Inherit VRF the VRF context is implicitly inherited for the ingress interface. Packets enter the VRF interface and are policy routed or forwarded out of the same VRF. The VRF routing and forwarding table is used when a route lookup is required to apply a set route policy to a packet.

- **Inter VRF**: For Inter VRF the VRF context needs to be specified explicitly. In this case, packets enter a VRF interface and are policy routed or forwarded to another VRF interface
- **VRF to Global Routing Table**: Packets enter the VRF interface and are policy routed or forwarded out of the Global Routing Table. The context for the Global Routing Table needs to be explicitly specified.
- **Global Routing Table to VRF**: Packets enter a global interface and are policy routed or forwarded out of a VRF interface

VRF aware PBR set clauses

You can enable VRF selection by PBR packets through one of the following options

- A route map
- The Global Routing Table
- A specified VRF

You can enable policy based routing of packets for a VRF instance by using route map commands with the following set clauses

- **set ip vrf *vrf-name* next-hop** *ip-address* [*ip-address*]: Indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set ipv6 vrf *vrf-name* next-hop** *ip-address* [*ip-address*]: Indicates where to route IPv6 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set global**: Routes the packets through the global routing table. The command is useful to route ingress packets belonging to a specific VRF through the global routing table.
- **set vrf**: Routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.
- **set ip global next-hop**: Indicates which next hop to forward the IPv4 packets that match the criterion of route-map for PBR. Uses the Global Routing table for reaching the next hop.
- **set ipv6 global next-hop**: Indicates which next hop to forward the IPv6 packets that match the criterion of route-map for PBR. Uses the Global Routing table for reaching the next hop.
- **set ip default vrf** *vrf-name* **nexthop** *ip-address* [*ip-address*]: Verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set ipv6 default vrf** *vrf-name* **nexthop** *ip-address* [*ip-address*]: Verifies the presence of the IPv6 address in the routing table of the VRF. If the IPv6 address is present the packet is not policy routed but forwarded based on the routing table. If the IPv6 address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set ip default global**: Configures IPv4 VRF to global routing.
- **set ipv6 default global**: Configures IPv6 VRF to global routing.
- **set ip default next-hop**: Indicates where to send IPv4 packets that pass a match criterion of a route map for PBR and for which no explicit route to a destination is specified.

- **set ipv6 default next-hop**: Indicates where to send IPv6 output packets that pass a match criterion of a route map for policy routing and for which no explicit route to a destination is specified.

How to Configure VRF aware PBR

Configuring Inherit-VRF in a Route Map

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ip access-list standard 10`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	Defines the criteria for which the access list will permit or deny packets.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_vrf1 permit 10`	Defines the conditions for enabling Policy Based Routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ip address 10`	Performs policy routing on matched packets. IP access lists and extended ACLs are supported.

	Command or Action	Purpose
Step 7	**match length min max** **Example:** `Device(config-route-map)# match length 64 1500`	Matches the length of the packet.
Step 8	**set ip next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip next-hop 135.35.35.2`	Specifies the next hop for routing packets.
Step 9	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 10	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 11	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 12	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)ip address 100.1.1.1 255.255.255.0`	Enters the IP address for the interface.
Step 13	**ip policy route-map** *map-tag* **Example:** `Device(config-if) ip policy route-map vrf1_vrf1`	Identifies the route map to use for PBR.
Step 14	**end** **Example:** `Device(config-f)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 15	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 16	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.

	Command or Action	Purpose
Step 17	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 18	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	Enters the IP address for the interface.

Configuring IPv6 Inherit-VRF in a Route Map

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ipv6 access-list acl_vrf1`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	Defines the criteria for which the access list will permit or deny packets.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_vrf1_v6 permit 10`	Defines the conditions for enabling Policy Based Routing. Enters route-map configuration mode.

	Command or Action	Purpose
Step 6	**match ip-address** { *acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ipv6 address acl_vrf1`	Performs policy routing on matched packets. IP access lists and extended ACLs are supported.
Step 7	**match length min max** **Example:** `Device(config-route-map)# match length 64 1500`	Matches the length of the packet.
Step 8	**set ip next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ipv6 next-hop 1335::1`	Specifies the next hop for IPv6 routing packets.
Step 9	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 10	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 11	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 12	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf) ipv6 address 1000::1/64`	Enters the IP address for the interface.
Step 13	**ip policy route-map** *map-tag* **Example:** `Device(config-if)ipv6 policy route-map vrf1_vrf1_v6`	Identifies the route map to use for PBR.
Step 14	**end** **Example:** `Device(config-if)end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 15	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 16	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 17	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 18	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf) ipv6 address 1335::2/64`	Enters the IP address for the interface.
Step 19	**ipv6 enable** **Example:** `Device(cofig-if) ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

Configuring Inter-VRF in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*]: Indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]: Verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set vrf**: Routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device# ip access-list standard 10`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_vrf2 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ip address 10`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • •IP access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.

Step 7	**set ip vrf** *vrf-name* **next-hop** {*ip-address* [*ip-address*] | }
	• **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*] | }
	• **set vrf***vrf-name*
	Example:
	`Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2`
	`or`
		The **set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*] command indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
		The **default** keyword verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the

	Command or Action	Purpose

	`Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2`
	or
	`Device(config-route-map)# set vrf vrf2`
		routing table, the packet is policy routed and sent to the specified next hop.
		The **set vrf** keyword routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 11	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0`	Enters the IP address for the interface.
Step 12	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map vrf1_vrf2`	Identifies the route map to use for PBR.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 14	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 15	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 16	**vrf forwarding** *vrf-name* **Example:**	Associates the VRF with the Layer 3 interface.

	Command or Action	Purpose
	`Device(config-if)# vrf forwarding vrf2`	
Step 17	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	Enters the IP address for the interface.

Configuring IPv6 Inter-VRF in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ipv6 vrf *vrf-name* next-hop***ip-address*[*ip-address*]: Indicates where to route IPv6 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]: Verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set vrf**: Routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device# ipv6 access-list acl_vrf1`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard*	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IPv6 addresses, IPv6 address ranges, and other IPv6 packet access

	Command or Action	Purpose
	Example: `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IPv6 access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ipv6 address acl_vrf1`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • •IPv6 access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.

Step 7	**set ip vrf** *vrf-name* **next-hop** {*ip-address* [*ip-address*] | }
	• **set ip default vrf***vrf-name* **next-hop** {*ip-address* [*ip-address*] | }
	• **set vrf***vrf-name*
	Example:
	`Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1`
	`or`
	`Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1`
	`or`
	`Device(config-route-map)# set vrf vrf2`
		The **set ipv6 vrf** ***vrf-name*** **next-hop** *ip-address*[*ip-address*] command indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
		The **default** keyword verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.

Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**vrf forwarding** *vrf-name* **Example:**	Associates the VRF with the Layer 3 interface.

	Command or Action	Purpose
	`Device(config-if)# vrf forwarding vrf1`	
Step 11	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)# ipv6 address 1000::1/64`	Enters the IP address for the interface.
Step 12	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ipv6 policy route-map vrf1_vrf2_v6`	Identifies the route map to use for PBR.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 14	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 15	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 16	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)vrf forwarding vrf2`	Associates the VRF with the Layer 3 interface.
Step 17	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf) ipv6 address 1335::2/64`	Enters the IP address for the interface.
Step 18	**ipv6 enable** **Example:** `Device(cofig-if) ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

Configuring VRF to Global Routing Table selection in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ip global next hop**: indicates where to forward IPv4/IPv6 packets that pass a match criterion of a route map for PBR and for which the global routing table is used.

- **set global**: routes the packets through the global routing table.

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device# ip access-list standard 10`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_global permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ip address 10`	Forwards any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • •IP access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.
Step 7	**set ip default global next-hop** *ip-address* [*ip-address*]	Specifies the next hop for routing packets.

	Command or Action	Purpose

	• **set global**
	Example:
	`Device(config-route-map)# set ip default global next-hop 135.35.35.2`
	`or`
	`Device(config-route-map)# set global`	

Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 11	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)#ip address 100.1.1.1 255.255.255.0`	Enters the IP address for the interface.
Step 12	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map vrf1_global`	Identifies the route map to use for PBR.
Step 13	**end** **Example:** `Device(config-f)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 14	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 15	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0`	Enters the IP address for the interface.

Configuring IPv6 VRF to Global Routing Table selection in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ipv6 global next hop**: indicates where to forward IPv6 packets that pass a match criterion of a route map for PBR and for which the global routing table is used.
- **set global**: routes the packets through the global routing table.

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device# ipv6 access-list acl_vrf1`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**}*protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map vrf1_global_v6 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] }	Forwards any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets.

	Command or Action	Purpose
	Example: `Device(config-route-map)# match ipv6 address acl_vrf1`	• •IP access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.

Step 7	**set ip default global next-hop** *ip-address* [*ip-address*]
	• **set global**
	Example:
	`Device(config-route-map)# set ipv6 default  global next-hop 1335::1`
	`or`
	`Device(config-route-map)# set global`
		Specifies the next hop for routing packets.

Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)vrf forwarding vrf1`	Associates the VRF with the Layer 3 interface.
Step 11	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf) ipv6 address 1000::1/64`	Enters the IP address for the interface.
Step 12	**ip policy route-map** *map-tag* **Example:** `Device(config-if)ipv6 policy route-map vrf1_global_v6`	Identifies the route map to use for PBR.
Step 13	**end** **Example:** `Device(config-if) end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 14	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.

	Command or Action	Purpose
Step 15	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf) ipv6 address 1335::2/64`	Enters the IP address for the interface.
Step 17	**ipv6 enable** **Example:** `Device(cofig-if) ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

Configuring Global Routing Table to VRF in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ip vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*]: Indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]: Verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set vrf**: Routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]	Specifies the IP access list type and enters the corresponding access list configuration mode.

	Command or Action	Purpose
	Example: `Device# ip access-list standard 10`	You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255`	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config-route-map)# route-map global_vrf permit 10`	Defines the conditions for forwarding routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ip address 10`	Forwards any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • •IP access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.

Step 7	**set ip vrf** *vrf-name* **next-hop** *ip-address* [*ip-address*]
	• **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*]
	• **set vrf***vrf-name*
	Example:
	`Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2`
	`or`
	`Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2`
	`or`
	`Device(config-route-map)# set vrf vrf2`
		The **set ip vrf** ***vrf-name*next-hop***ip-address*[*ip-address*] command indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
		The **default** keyword verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
		The **set vrf** keyword routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

	Command or Action	Purpose
Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)ip address 100.1.1.1 255.255.255.0`	Enters the IP address for the interface.
Step 11	**ip policy route-map** *map-tag* **Example:** `Device(config-if) ip policy route-map global_vrf1`	Identifies the route map to use for PBR.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 14	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 15	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf2`	Associates the VRF with the Layer 3 interface.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0`	Enters the IP address for the interface.

Configuring IPv6 Global Routing Table to VRF in a Route Map

Before you begin

You can use the following set clauses of the route-map commands:

- **set ipv6 vrf *vrf-name* next-hop***ip-address*[*ip-address*]: Indicates where to route IPv6 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
- **set ip default vrf** *vrf-name***nexthop***ip-address*[*ip-address*]: Verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
- **set vrf**: Routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

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 access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device# ipv6 access-list acl_vrf1`	Specifies the IP access list type and enters the corresponding access list configuration mode. You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* **Example:** `Device(config-ipv6-acl)# 10 permit ipv6 1333::/64 2000::/64`	Defines the criteria for which the access list will permit or deny packets. Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options in Cisco IOS software to define match criteria.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:**	Defines the conditions for forwarding routes from one routing protocol into another, or enables policy routing. Enters route-map configuration mode.

	Command or Action	Purpose
	`Device(config-route-map)# route-map global_vrf_v6 permit 10`	
Step 6	**match ip-address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] } **Example:** `Device(config-route-map)# match ipv6 address acl_vrf1`	Forwards any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • •IPv6 access lists are supported. • •The example configures the route map to use standard access list 1 to define match criteria.

Step 7	**set ip vrf** *vrf-name* **next-hop** *ip-address* [*ip-address*]
	• **set ip default vrf***vrf-name***next-hop** {*ip-address* [*ip-address*]
	• **set vrf***vrf-name*
	Example:
	`Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1`
	or
	`Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1`
	or
	`Device(config-route-map)# set vrf vrf2`
		The **set ipv6 vrf** ***vrf-name*****next-hop***ip-address*[*ip-address*] command indicates where to route IPv4 packets that pass a match criteria of a route map using the next-hop specified for the VRF.
		The **default** keyword verifies the presence of the IP address in the routing table of the VRF. If the IP address is present the packet is not policy routed but forwarded based on the routing table. If the IP address is absent in the routing table, the packet is policy routed and sent to the specified next hop.
		The **set vrf** keyword routes packets using a particular VRF table through any of the interfaces belonging to that VRF. If there is no route in the VRF table, the packet will be dropped.

Step 8	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config-if)# interface HundredGigE1/0/11`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 9	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 10	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)# ipv6 address 1000::1/64`	Enters the IP address for the interface.
Step 11	**ip policy route-map** *map-tag* **Example:**	Identifies the route map to use for PBR.

	Command or Action	Purpose
	`Device(config-if)# ipv6 policy route-map global_vrf_v6`	
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	**interface HundredGigE** *rack/slot/module/port* **Example:** `Device(config)# interface HundredGigE1/0/25`	Configures a Hundred Gigabit Ethernet interface and enters interface configuration mode.
Step 14	**no switchport** **Example:** `Device(config-if)# no switchport`	Configures the interface as a Layer 3 Ethernet interface.
Step 15	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf2`	Associates the VRF with the Layer 3 interface.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if-vrf)# ipv6 address 1335::2/64`	Enters the IP address for the interface.
Step 17	**ipv6 enable** **Example:** `Device(cofig-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

Configuration Examples for VRF aware PBR

Example: Configuring a VRF interface as an inherit VRF in a route map

This example shows how to configure a VRF interface as a inherit VRF in a route map.

```
Device(config)# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf1 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# match length 64 1500
Device(config-route-map)#  set ip next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_vrf1
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
```

```
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring an IPv6 VRF interface as an inherit VRF in a route map

This example shows how to configure an IPv6 VRF interface as a inherit VRF in a route map.

```
Device(config)# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf1_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# match length 64 1500
Device(config-route-map)# set ipv6  next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_vrf1_v6

Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring a VRF interface as an Inter VRF in a route map using the set ip vrf clause

This example shows how to configure a VRF interface as an Inter VRF in a route map using the **set ip vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip  vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_vrf1
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring a VRF interface as an IPv6 Inter VRF in a route map using the set ip vrf clause

This example shows how to configure an IPv6 VRF interface as an Inter VRF in a route map using the **set ip vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
```

```
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_vrf1_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(cofig-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring a VRF interface as an Inter VRF in a route map using the set ip default vrf clause

This example shows how to configure a VRF interface as an Inter VRF in a route map using the **set ip vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)#ip address 100.1.1.1 255.255.255.0
Device(config-if-vrf)# ip policy route-map vrf1_vrf2
Device(config-if-vrf)# end
Device(config-if)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring an IPv6 VRF interface as an Inter VRF in a route map using the set ip default vrf clause

This example shows how to configure an IPv6 VRF interface as an Inter VRF in a route map using the **set ip vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv6-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
Device(config-route-map)#  match ipv6 address acl_vrf1
Device(config-route-map)#  set ipv6  default vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if-vrf)# ipv6 policy route-map vrf1_vrf2_v6
Device(config-if-vrf)# end
Device(config-if)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring a VRF interface as an Inter VRF in a route map using the set vrf clause

This example shows how to configure a VRF interface as an Inter VRF in a route map using the **set vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_vrf2 permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_vrf2
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring an IPv6 VRF interface as an Inter VRF in a route map using the set vrf clause

This example shows how to configure an IPv6 VRF interface as an Inter VRF in a route map using the **set vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_vrf2_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(confog-f)# ipv6 policy route-map vrf1_vrf2_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring a VRF to Global Routing Table in a Route Map using the set ip default global clause

This example shows how to configure packets from a VRF to Global Routing Table in a route map using the **set ip default global** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_global permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default global next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
```

```
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_global
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring an IPv6 VRF to Global Routing Table in a Route Map using the set ip default global clause

This example shows how to configure packets from an IPv6 VRF to Global Routing Table in a route map using the **set ip default global** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)#  sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_global_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)#  set ipv6 default  global next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_global_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring a VRF to Global Routing Table in a Route Map using the set global clause

This example shows how to configure packets from a VRF to Global Routing Table in a route map using the **set global** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map vrf1_global permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set global
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map vrf1_global
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)ip address 135.35.35.1 255.255.255.0
```

Example: Configuring an IPv6 VRF to Global Routing Table in a Route Map using the set global clause

This example shows how to configure packets from an IPv6 VRF to Global Routing Table in a route map using the **set global** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv6-acl)#  sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map vrf1_global_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set global
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf1
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if)# ipv6 policy route-map vrf1_global_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring Global Routing Table to VRF in a Route Map using the set ip vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to a VRF in a route map using the **set ip vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map global_vrf
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring Global Routing Table to an IPv6 VRF in a Route Map using the set ipv6 vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to an IPv6 VRF in a route map using the **set ipv6 vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ipv6 address 1000::1/64
Device(config-if)#  ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring Global Routing Table to VRF in a Route Map using the set ip default vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to a VRF in a route map using the **set ip vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set ip default vrf vrf2 next-hop 135.35.35.2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if-vrf)#ip policy route-map global_vrf
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ip address 135.35.35.1 255.255.255.0
```

Example: Configuring Global Routing Table to IPv6 VRF in a Route Map using the set ipv6 default vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to a VRF in a route map using the **set ipv6 default vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set ipv6 default vrf vrf2 next-hop 1335::1
Device(config-if)# interface HundredGigE1/0/11
Device(config-if-vrf)#  ipv6 address 1000::1/64
Device(config-if-vrf)# ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)# ipv6 address 1335::2/64
Device(config-if-vrf)# ipv6 enable
```

Example: Configuring Global Routing Table to VRF in a Route Map using the set vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to a VRF in a route map using the **set vrf** clause.

```
Device# ip access-list standard 10
Device(config-ipv4-acl)# 10 permit 133.33.33.0 0.0.0.255
Device(config-route-map)# route-map global_vrf permit 10
Device(config-route-map)# match ip address 10
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip policy route-map global_vrf
Device(config-if)# end
```

```
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)#ip address 135.35.35.1 255.255.255.0
```

Example: Configuring Global Routing Table to IPv6 VRF in a Route Map using the set vrf clause

This example shows how to configure routing and forwarding of packets from Gloal Routing Table to an IPv6 VRF in a route map using the **set vrf** clause.

```
Device# ipv6 access-list acl_vrf1
Device(config-ipv4-acl)# sequence 10 permit ipv6 1333::/64 2000::/64
Device(config-route-map)# route-map global_vrf_v6 permit 10
Device(config-route-map)# match ipv6 address acl_vrf1
Device(config-route-map)# set vrf vrf2
Device(config-if)# interface HundredGigE1/0/11
Device(config-if)# no switchport
Device(config-if-vrf)#  ipv6 address 1000::1/64
Device(config-if)#  ipv6 policy route-map global_vrf_v6
Device(config-if)# end
Device(config)# interface HundredGigE1/0/25
Device(config-if)# no switchport
Device(config-if)# vrf forwarding vrf2
Device(config-if-vrf)#ipv6 address 1335::2/64
Device(config-if-verf)# ipv6 enable
```

Feature History for VRF aware PBR

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 Gibraltar 16.12.1	VRF aware PBR	PBR can be configured on VRF lite interfaces.

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/.