Configuring Policy-Based Routing

This chapter describes the tasks for configuring policy-based routing (PBR) on a Catalyst 4500 series switch and includes these major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, see the Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch.

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference* , you can locate it in the Cisco IOS Master Command List, All Releases.

Note To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Policy-Based Routing

Policy-Based Routing (PBR) gives you a flexible method of routing packets by allowing you to define policies for traffic flows, lessening reliance on routes derived from routing protocols. PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to specify paths for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, or an application protocol. Enable PBR to provide the following advantages:

- Equal access
- Protocol-sensitive routing
- Source-sensitive routing
- Routing based on interactive versus batch traffic
- Routing based on dedicated links

Some applications or traffic can benefit from source-specific routing; for example, you can transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data, such as e-mail, over a lower-bandwidth, lower-cost link

Policies can be based on IP address, port numbers, or protocols. For a simple policy, use any one of these descriptors; for a complicated policy, all of them.

Route Maps

The following topics are discussed in the section:

Understanding Route Maps

All packets received on an interface with PBR enabled (except those sent directly to the switch IP) are handled by enhanced packet filters known as route maps. The route maps dictate the policy that determines where the packets are forwarded.

Route maps contain statements that can be marked as permit or deny. They are interpreted in the following ways:

- If a statement is marked as deny, the packets meeting the match criteria are sent back using the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and a packet matches the access-lists, then the first valid set clause is applied to that packet.

You can implement PBR by applying a route map on an incoming interface. A given interface can have only one route-map configured. A route map is configured at the global configuration parser mode. You can then apply this route map on one or more interfaces (in the interface configuration parser sub-mode).

Each route map statement contains **match** and **set** commands. The **match** command denotes the match criteria to be applied on the packet data. The **set** command denotes the PBR action to be taken on the packet.

The following example shows a single route map called rm-test and six route map statements:

```
    route-map rm-test permit 21
     match ip address 101
     set ip next-hop 21.1.1.1
    !
    route-map rm-test permit 22
     match ip address 102
     set ip next-hop 22.2.2.1
    !
    route-map rm-test permit 23
     match ip address 101 2102
     set interface vlan23
!
    route-map rm-test deny 24
     match ip address 104
     set ip next-hop 24.4.4.1
```

```
!
route-map rm-test deny 25
 match ip address 105
 set ip next-hop 25.5.5.1
!
route-map rm-test permit 26
 match ip address 2104
 set ip next-hop 26.6.6.1
```

The numbers 21, 22, ... 26 are the sequence numbers of the route-map statements.

PBR Route-Map Processing Logic

When a packet is received on an interface configured with a route map, the forwarding logic processes each route map statement according to the sequence number.

If the route map statement encountered is a **route-map... permit** statement:

- The packet is matched against the criteria in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the decision reached is permit, then the PBR logic executes the action specified by the **set** command on the packet.
- If the decision reached is deny, then the PBR action (specified in the **set** command) is not applied. Instead the processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

If the route map statement encountered is a **route-map... deny** statement:

- The packet is matched against the criteria given in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the criteria decision is permit, then the PBR processing terminates, and the packet is routed using the default IP routing table.
- If the criteria decision is deny, then the PBR processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

Note The **set** command has no effect inside a **route-map... deny** statement.

A route map statement can have multiple **set** commands that are applied in the following priority:

set ip next-hop verify-availability

set ip next-hop

set ip next-hop recursive

set interface

set default ip next-hop

set default interface

If both the **set ip next-hop** and **set ip next-hop recursive** commands are present in the same route-map statement, the **next-hop set** command is applied.

If the **set ip next-hop** command is not available then the **set ip next-hop recursive** command is applied.

If the **set ip recursive-next-hop** and the **set** interface command are not present, then the packet is routed using the default routing table; it is not dropped. If the packet is required to be dropped, use the **set next-hop recursive** command followed by a **set interface null0 configuration** command.

Load Balancing with Recursive Next Hop

If multiple equal-cost routes to the subnet have been configured by the **set ip next-hop recursive** command, load balancing will occur only if all the adjacencies to the routes are resolved. If any of the adjacencies have not been resolved, then load balancing will not happen and only one of the routes whose adjacency is resolved will be used. If none of the adjacencies are resolved, then packets will be processed in software, resulting in at least one of the adjacencies to be resolved and programmed in hardware. PBR relies on routing protocols or other means to resolve all adjacencies and make load balancing happen.

Packet Matching Criteria

Access Control Lists (ACLs) define the allowed match criteria for packets. Each ACL is applied to incoming packets in a certain order, stopping only when the packet characteristics match the ACL being applied. Unlike policy maps, route maps do not support the "match any" match semantics.

IPv6 packets are matched via a **match ipv6 address** statement in the associated PBR route-map. IPv6 PBR requires IPv6 ACL, although the statement may specify either an IPv6 ACL or an IPv6 Prefixlist,

Packets are matched using the following criteria:

- Input interface
- Source IPv4/IPv6 Address (Prefixlist/Standard/Extended ACL)
- Destination IPv4/IPv6 Address (Standard/Extended ACL)
- Protocol (Extended ACL)
- Source Port and Destination Port (Extended ACL)
- DSCP (Extended ACL)
- Flow-label (Extended ACL)
- Fragment (Extended ACL)

PBR Route-Map Processing Logic Example

Consider a route map called rm-test defined as follows:

```
access-list 101  permit tcp host 61.1.1.1 host 133.3.3.1 eq 101
access-list 102  deny   tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 2102 permit tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 104  deny   tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 2104 permit tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 105  permit tcp host 61.1.1.1 host 133.3.3.1 eq 105


route-map rm-test permit 21
 match ip address 101
 set ip next-hop 21.1.1.1
!
route-map rm-test permit 22
 match ip address 102
 set ip next-hop 22.2.2.1
```

```
 !
 route-map rm-test permit 23
  match ip address 101 2102
  set interface vlan23
 !
 route-map rm-test deny 24
  match ip address 104
  set ip next-hop 24.4.4.1
!

 route-map rm-test deny 25
  match ip address 105
  set ip next-hop 25.5.5.1
 !
 route-map rm-test permit 26
  match ip address 2104
  set ip next-hop 26.6.6.1
```

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 101
  - Matches ACL 101 in sequence #21.
  - PBR is switched through next-hop 21.1.1.1.

Note ACL 101 is also matched in sequence #23, but the processing doesn't reach that point

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 102
  - In sequence #21, the ACL 101 action denies this packet (because all ACLs have an implicit deny). Processing advances to sequence #22.
  - In sequence #22, ACL 102 matches TCP port 102, but the ACL action is deny. Processing advances to sequence #23.
  - In sequence #23, ACL 2102 matches TCP port 102, and the ACL action is permit.
  - Packet is switched to output interface VLAN 23.
- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 105
  - Processing moves from sequence #21 to #24, because all ACLs in these sequence numbers have a deny action for port 105.
  - In sequence #25, ACL 105 has a permit action for TCP port 105.
  - The route-map deny command takes effect, and the packet is routed using the default IP routing table.

The Catalyst 4500 series switch supports matching route map actions with a packet by installing entries in the TCAM that match the set of packets described by the ACLs in the match criteria of the route map. These TCAM entries point at adjacencies that either perform the necessary output actions or forward the packet to software if either hardware does not support the action or its resources are exhausted.

If the route map specifies a **set interface …** action, packets that match the **match** statement are routed in the software. Some packets may be dropped. Similarly, if the route-map specifies a **set default interface…** action and there is no matching IP route for the packet, the packet is routed in the software.

Note The scale of hardware-based PBR is determined by the TCAM size and the time required for the CPU to flatten the ACL before programming into the hardware. The time take to flatten the ACL increases when a PBR policy requires a considerable number of route-maps. For example, a PBR policy of 1,200

route-maps (each containing ACLs with permit ACEs only) may require 6-7 minutes of flatten time before programming into hardware. This process may repeat if an adjacency change requires PBR reprogramming.

Policy-Based Routing with Object Tracking

Beginning in Cisco IOS XE Release 3.8.0E and Cisco IOS Release 15.2(4)E, you can configure Policy-Based Routing (PBR) to use object tracking, to verify the most viable next-hop IP address to which to forward packets, using an Internet Control Message Protocol (ICMP) ping as the verification method. PBR used with object tracking is most suitable for devices that have multiple Ethernet connections as the next hop. Normally, Ethernet interfaces connect to digital subscriber line (DSL) modems or cable modems, and do not detect a failure upstream in the ISP broadband network. The Ethernet interface remains up, and any form of static routing points to that interface. Using PBR with object tracking allows you to back-up two Ethernet interfaces, determine the interface that is available by sending ICMP pings to verify if the IP address can be reached, and then route traffic to that interface.

To verify the next-hop IP address for the device, PBR informs the object tracking process that it is interested in tracking a certain object. The tracking process, in turn, informs PBR when the state of the object changes.

Restrictions for Policy-Based Routing with Object Tracking

The **set next-hop verify-availability** command is not supported with the following:

- VRF instances
- Virtual switching system (VSS)
- IPv6 traffic

IPv4 and IPv6 Policy-Based Routing for VRF Instances

Virtual routing and forwarding (VRF) allows multiple routing instances in Cisco software. Beginning in Cisco IOS Release XE 3.7 0E and IOS 15.2(3)E, the Policy-Based Routing (PBR) feature is VRF-aware and works on multiple routing instances, beyond the default or global routing table.

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set** command configuration determines the VRF through which outbound packets are policy routed.

Inherit-VRF, Inter-VRF, Global-to-VRF, and VRF-to-Global Routing

The Policy-Based Routing feature supports inherit-VRF, inter-VRF, and VRF-to-global routing.

With inherit-VRF, packets arriving at a virtual routing and forwarding (VRF) interface are routed, by looking-up the same VRF's routing table.

With inter-VRF routing, packets arriving at a VRF interface are routed, by looking-up a different VRF's routing table, as specified by the **set** command.

With VRF-to-global routing, packets arriving at a VRF interface are routed via the global routing table.

With global-to-VRF routing, packets arriving at the global interface (an interface that is not part of a VRF) are routed via a VRF routing table.

Restrictions for VRF-Aware Policy-Based Routing

- The same route-map cannot be used to configure PBR:
  - on interfaces that belong to different VRFs
  - on one VRF interface and another global interface (an interface that is not part of a VRF).
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface**, s**et ip default next-hop**, and **set ip next-hop commands**. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface**, **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three set commands.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- When you use the **set vrf** command you specify the VRF table to be looked-up; this overrides the default or global routing table. If a route is not specified in the VRF routing table, then packets are dropped (even if a route exists in the global routing table).
- The **set next-hop verify-availability** and the **set ip next hop recursive** commands are not supported within VRF instances.

Policy-Based Routing Configuration Tasks

To configure PBR, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional. For configuration examples, see the "Policy-Based Routing Configuration Examples" section on page 40-16.

Enabling IPv4 PBR

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must apply that route-map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

To enable IPv4 PBR on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	`Switch(config-route-map)# match ip address {access-list-number | name} [...access-list-number | name]`	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended IP access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.
Step 3	`Switch(config-route-map)# set ip next-hop ip-address [... ip-address]`	Specifies the next-hop IP address to which matching packets are sent. The next-hop IP address specified here must belong to a subnet that is directly connected to this switch. If more than one next-hop IP address is specified, the first usable next-hop is chosen for routing matching packets. If the next-hop is (or becomes) unavailable for some reason, the next one in the list is chosen.
Step 4	`Switch(config-route-map)# set ip next-hop verify-availability [next-hop-address-sequence track object]`	(Optional) Configures the route map to verify the reachability of the tracked object. **Note** This option is not supported for IPv6 traffic. For information about defining new tracked object, see Verifying Next-Hop IP using Object Tracking , page 40-14
Step 5	`Switch(config-route-map)# set ip next-hop recursive ip-address`	Specifies a recursive next-hop IP address. **Note** The recursive next-hop can be a subnet that is not directly connected. The **set ip next-hop recursive** command does not ensure that packets are routed through the recursive-next-hop if there is an intermediate node with a shorter route to the destination such that the route does not pass through the recursive-next-hop.
Step 6	`Switch(config-route-map)# set interface interface-type interface-number [... type number]`	Specifies the output interface from which the packet will be sent. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (not a switchport). Packets are forwarded on the specified interface only if one of the following conditions is met: • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.k

	Command	Purpose
Step 7	`Switch(config-route-map)# set ip default next-hop ip-address [... ip-address]`	Sets next hop to which to route the packet if there is no explicit route for the destination IP address in the packet. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.
Step 8	`Switch(config-route-map)# set default interface interface-type interface-number [...type ...number]`	Specifies the output interface from which the packet will be sent if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by using the routing table. If no match is found, the packet is forwarded to the specified output interface. Packets are forwarded on the specified interface only if one of the following conditions is met: • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.
Step 9	`Switch(config-route-map)# interface interface-type interface-number`	Specifies the interface. This command puts the switch into interface configuration mode.
Step 10	`Switch(config-if)# ip policy route-map map-tag`	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Use the **set** commands with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local switch finds a next hop and a usable interface, it routes the packet.

Refer to the section Policy-Based Routing Configuration Examples, page 40-16 for examples of IPv4 PBR.

Use the **show route-map map-tag** command to display the existing route map.

Note Packet and byte counters in the output of the **show route-map map-tag** command are not updated.

Enabling IPv6 PBR

Note With IOS XE 3.6.0E and IOS 15.2(2)E, IPv6 PBR is not supported on Supervisor Engine 8-E.

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must apply that route-map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

For IPv6 PBR to work on the Catalyst 4900M, Catalyst 4948E and Catalyst 4948E-F Series Switches, IPv4 and IPv6 routing must be enabled the device.

To enable IPv6 PBR on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	`Switch(config-route-map)# match ipv6 address {access-list-number | name} [...access-list-number | name]`	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended ipv6 access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.
Step 3	`Switch(config-route-map)# set ipv6 next-hop ip-address [... ip-address]`	Specifies the next-hop IP address to which matching packets are sent. The next-hop IP address specified here must belong to a subnet that is directly connected to this switch. If more than one next-hop IP address is specified, the first usable next-hop is chosen for routing matching packets. If the next-hop is (or becomes) unavailable for some reason, the next one in the list is chosen.
Step 4	`Switch(config-route-map)# set interface interface-type interface-number [... type number]`	Specifies the output interface from which the packet will be sent. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (not a switchport). Packets are forwarded on the specified interface only if one of the following conditions is met: • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.k

	Command	Purpose
Step 5	`Switch(config-route-map)# set ipv6 default next-hop ip-address [... ip-address]`	Sets next hop to which to route the packet if there is no explicit route for the destination IP address in the packet. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.
Step 6	`Switch(config-route-map)# set default interface interface-type interface-number [...type ...number]`	Specifies the output interface from which the packet will be sent if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by using the routing table. If no match is found, the packet is forwarded to the specified output interface. Packets are forwarded on the specified interface only if one of the following conditions is met: • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.
Step 7	`Switch(config-route-map)# interface interface-type interface-number`	Specifies the interface. This command puts the switch into interface configuration mode.
Step 8	`Switch(config-if)# ipv6 policy route-map map-tag`	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Note The **recursive** option is supported for IPv4, but not for IPv6. An interface can have either an ipv4 route map or an ipv6 route map. An interface can be bound to only one route map.

Use the **set** commands with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local switch finds a next hop and a usable interface, it routes the packet.

Refer to the following document for IPv6 PBR configuration examples.

http://www.cisco.com/c/en/us/support/docs/ip/ip-version-6-ipv6/112218-policy-based-routing-ipv6-configex.html

Note Packet and byte counters in the output of the **show route-map map-tag** command are updated only for software switched packets. Counters for hardware switched packets are not updated.

Enabling Local IPv4 and Local IPv6 PBR

Packets that are generated by the switch are not normally policy-routed. To enable local PBR for such packets, indicate which route map the switch should use by entering this command:

IPv4

Command	Purpose
`Switch(config)# ip local policy route-map map-tag`	Identifies the IPv4 route map to use for local PBR.

IPv6

Command	Purpose
`Switch(config)# ipv6 local policy route-map map-tag`	Identifies the IPv6 route map to use for local PBR.

All packets originating on the switch are then subject to local PBR.

Use the **show ip local policy** command to display the route map used for local PBR, if one exists.

Configuring IPv4 and IPv6 PBR for VRF Instances

To enable PBR for multiple routing instances, configure your device in the following way:

	Command	Purpose
Step 1	`Switch(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.

Step 2	For IPv4:
	`Switch(config-route-map)# match ip address {access-list-number | name} [...access-list-number | name]`
	For IPv6:
	`Switch(config-route-map)# match ipv6 address {access-list-number | name} [...access-list-number | name]`
		Specifies the match criteria. The match criteria take the form of one or more Standard or Extended access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.

	Command	Purpose

Step 3	Set one of the following
	For IPv4:
	`Switch(config-route-map)# set ip vrf [vrf name] next-hop ip address> [...ip-address]`
	For IPv6:
	`Switch(config-route-map)# set ipv6 vrf [vrf name] next-hop ip address> [...ip-address]`
		Specifies the next-hop IP address under the VRF, to which the matched packets must be forwarded.The next-hop IP address must exist in the routing table, under the VRF.

	For IPv4:
	`Switch(config-route-map)# set ip global next-hop ip address> [...ip-address]`
	For IPv6:
	`Switch(config-route-map)# set ipv6 global next-hop ip address> [...ip-address]`
		Specifies the next-hop IP address, from the global routing table, to which to forward matched packets. The next-hop IP address must exist in the global routing table.

Step 4	Set one of the following:
	For IPv4:
	`Switch(config-route-map)# set ip default vrf [vrf name] next-hop ip address> [...ip-address]`
	For IPv6:
	`Switch(config-route-map)# set ipv6 default vrf [vrf name] next-hop ip address> [...ip-address]`
		Specifies the next-hop IP address to which the matched packets must be forwarded when there is no explicit packet destination address in the routing table, under the VRF.

	For IPv4:
	`Switch(config-route-map)# set ip default global next-hop ip address> [...ip-address]`
	For IPv6:
	`Switch(config-route-map)# set ipv6 default global next-hop ip address> [...ip-address]`
		Specifies the next-hop IP address to which the matched packets must be forwarded when there is no explicit packet destination address corresponding to the VRF to which the interface belongs, in the routing table. The next-hop address specified must exist in the global routing table.

	Command	Purpose

Step 5	Set one of the following:
	For global routing:
	`Switch(config-route-map)# set global`
	For inter-VRF routing:s
	`Switch(config-route-map)# set vrf [vrf name]`
		Specifies that the global routing table should be looked-up to route packets,
		Use the **set global** command to configure VRF-to-Global routing.
		Use the **set vrf** command to specify the VRF table to be looked-up, to route packets.
		Use this command to configure Inter-VRF routing and route packets arriving at a particular VRF interface through a different VRF interface, by looking-up a different VRF's routing table. Using this command overrides the default or global routing table. If a route is not specified in the VRF routing table, then packets are dropped (even if a route exists in the global routing table).

Step 6	`Switch(config-route-map)# interface` *interface-type interface-number*	Specifies the interface. This command puts the switch into interface configuration mode.
Step 7	`Switch(config-if)# ipv6 policy route-map` *map-tag*	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Verifying the PBR Configuration for VRF Instances

To verify the PBR configuration for VRF instances, enter the following steps in any order:

	Command	Purpose
Step 1	`Switch# show ip access list` [*access-list-number* | *access-list-name*]	Displays the subnet ranges defined as match criteria in the standard access lists.
Step 2	`Switch# show route-map` [*map-name*]	Displays the match and set commands in the route map.

The following example shows you how to configure PBR for VRF instances:

```
Switch# enable
Switch# configure terminal
Switch(config)# route-map map1 permit 10
Switch(config-route-map)# set ipv6 vrf myvrf next-hop 2001.DB8:4:1::1/64
Switch(config-route-map)# end
Switch# show route-map map1
```

Verifying Next-Hop IP using Object Tracking

To verify the next-hop IP address using PBR with Object Tracking, perform the following steps:

Note The **set ip next-hop verify-availability** command is not supported on VRF instances, on a virtual switching system (VSS), and with IPv6 traffic.

	Command	Purpose
Step 1	`Switch# ocnfigure terminal`	Enters global configuration mode.
Step 2	`Switch (config)# track [object-number] ip sla [entry-number]`	Tracks the state of the specified IP SLA object.
Step 3	`Switch (config)#ip sla [operation-number]`	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA configuration mode.
Step 4	`Switch (config-ip-sla)# icmp echo [ip-address] source ip [ip-address]`	Configures an IP SLA Internet Control Message Protocol (ICMP) echo probe operation and enters Echo configuration mode.
Step 5	`Switch (config-ip-sla-echo)# frequency seconds`	(Optional) Sets the rate at which a specified IP SLA operation repeats.
Step 6	`Switch (config-ip-sla-echo)# threshold milliseconds`	(Optional) Sets the length of time required for a rising threshold event to be declared.
Step 7	`Switch (config-ip-sla-echo)# timeout milliseconds`	(Optional) Sets the maximum time required for the IP SLA operation to be completed.
Step 8	`Switch (config)#ip sla schedule [operation-number] [life {forever | seconds}] [start-time {hh : mm} [: ss] [monthday | daymonth]| pending | now | after hh : mm : ss}] [ageout seconds]`	Configures the scheduling parameters for a single Cisco IOS IP SLA operation.
Step 9	`Switch(config)# route-map map-tag [permit | deny] [sequence-number]`	Specifies a route map and enters route-map configuration mode.
Step 10	`Switch(config-route-map)# match ip address [access-list-name]`	Distributes routes that have a destination IPv4 network number address that is permitted by a standard access list.
Step 11	`Switch(config-route-map)# set ip next-hop verify-availability [next-hop-address sequence track object]`	Configures the route map to verify the reachability of the tracked object.

The following example shows you how to verify the next-hop IP address in a route map:

```
Switch# enable
Switch# configure terminal
Switch(config)# track 100 ip sla 100
Switch(config)# ip sla 100
switch(config-ip-sla)# icmp-echo 172.19.255.253 source-ip 172.19.255.47
switch(config-ip-sla-echo)# timeout 1500
switch(config-ip-sla-echo)# threshold 1000
switch(config-ip-sla-echo)# frequency 2
switch(config)# ip sla schedule 100 life forever start-time now
switch(config)# route-map alpha permit 10
switch(config-route-map)# match ip address exlist
switch(config-route-map)# set ip next-hop verify-availability 95.1.1.2 1 track 100
switch# show route-map alpha
switch# show track 100
```

Unsupported Commands

The following PBR commands in config-route-map mode are in the CLI but not supported in Cisco IOS for the Catalyst 4500 series switches. If you attempt to use these commands, an error message displays:

- **match-length**
- **set ip qos** and **set ipv6 qos**
- **set ip tos** and **set ipv6 tos**
- **set ip precedence** and **set ipv6 precedence**
- **set ip df** and **set ipv6 df**
- **set ipv6 next-hop recursive**
- **set ipv6 next-hop verify-availability**

Policy-Based Routing Configuration Examples

The following sections provide PBR configuration examples:

- Equal Access, page 40-16
- Differing Next Hops, page 40-17
- Deny ACE, page 40-17

For information on how to configure policy-based routing, see the section "Policy-Based Routing Configuration Tasks" in this chapter.

Equal Access

The following example provides two sources with equal access to two different service providers. Packets arriving on interface fastethernet 3/1 from the source 10.1.1.1 are sent to the switch at 6.6.6.6 if the switch has no explicit route for the destination of the packet. Packets arriving from the source 2.2.2.2 are sent to the switch at 7.7.7.7 if the switch has no explicit route for the destination of the packet. All other packets for which the switch has no explicit route to the destination are discarded.

```
Switch (config)# access-list 1 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map equal-access
!

route-map equal-access permit 10
 match ip address 1
 set ip default next-hop 6.6.6.6
route-map equal-access permit 20
 match ip address 2
 set ip default next-hop 7.7.7.7
route-map equal-access permit 30
 set default interface null0
```

Note If the packets you want to drop do not match either of the first two route-map clauses, then change **set default interface null0** to **set interface null0**.

Differing Next Hops

The following example illustrates how to route traffic from different sources to different places (next hops). Packets arriving from source 10.1.1.1 are sent to the next hop at 3.3.3.3; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5.

```
access-list 1 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
 set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
 set ip next-hop 3.3.3.5
```

Deny ACE

The following example illustrates how to stop processing a given route map sequence, and to jump to the next sequence. Packets arriving from source 10.1.1.1 skip sequence 10 and jump to sequence 20. All other packets from subnet 10.1.1.0 follow the set statement in sequence 10.

```
access-list 1 deny ip 10.1.1.1
access-list 1 permit ip 10.1.1.0 0.0.0.255
access-list 2 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
ip policy route-map Texas
!
route-map Texas permit 10
match ip address 1
set ip next-hop 3.3.3.3
!
route-map Texas permit 20
match ip address 2
set ip next-hop 3.3.3.5
```

Examples of the show Command

The following **show** command illustrates that route map pbrv6-test has only one permit sequence.

In the example policy, IPv6 packets with an address matching the criteria defined by the access control list v6_acl are forwarded to the next hop 2006::2. If next-hop 2006::2 is unreachable, the matching packets are forwarded to 2005::2. If both next-hops are unreachable, the packets are forwarded using the routing table lookup. For packets that do not match the filter criteria, a standard routing table lookup is performed for packet forwarding.

```
Switch# show route-map pbrv6-test
route-map pbrv6-test, permit, sequence 10
  Match clauses:
     ipv6 address v6_acl
  Set clauses:
     ipv6 next-hop 2006::2 2005::2
```

```
Policy routing matches: 0 packets, 0 bytes
```