Routing Policies Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes how to configure routing policies on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

Routing Policy Overview

Routing policy

A routing policy is a mechanism that allows a user to configure criteria to compare a routing update against, with one or more actions to be performed on the route if the defined criteria are met. For example, a policy can be created to filter (block) specific route prefixes that are being announced by a BGP neighbor. Policy statements are also used to export routes learned via one protocol, for instance OSPF, into another protocol, for instance BGP. This is commonly called route redistribution.

Routing policies are grouped together in the AT&T Vyatta vRouter configuration under the **`policy`** node. This **`policy`** node simply serves as a container for policy statements; it's the actual policy statements that define the rules that will be applied to routing updates.

Once a policy has been defined, in order for it to take affect, it needs to be applied to a specific routing protocol. A policy can be applied as either an import policy or an export policy to routing protocols like RIP, OSPF, and BGP. In the case of BGP, policies can be applied per peer. Only one import and one export policy can be applied to a protocol (or a BGP peer).

A policy that has been applied as an import policy to a routing protocol is used to evaluate routing updates received through the routing protocol to which the policy is applied. For example, if a user configures an import policy for the BGP protocol, all BGP announcements received by the AT&T Vyatta vRouter is compared against the import policy first, prior to being added to the BGP and routing tables.

A policy that has been applied as an export policy to a routing protocol is used to evaluate routing updates that are transmitted by the routing protocol to which the policy is applied. For example, if a user configures an export policy for BGP, all BGP updates originated by the AT&T Vyatta vRouter will be compared against the export policy statement prior to the routing updates being sent to any BGP peers.

In addition to controlling routing updates transmitted by a routing protocol, export policies are also used to provide route redistribution. For example, if a user wants to redistribute routes learned through OSPF into BGP, the user would configure a policy statement identifying the OSPF routes of interest, and then the user would apply this policy statement as an export policy for OSPF.

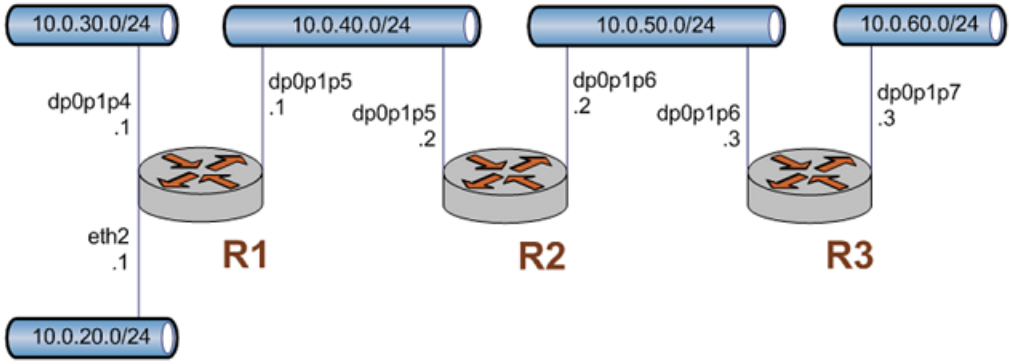
Routing Policy Configuration Examples

Filtering routes using access lists

Access lists can be used to filter routes for distance-vector protocols such as RIP and at redistribution points into link-state routing domains (like OSPF) where they can control which routes enter or leave the domain.

This section presents a sample configuration for RIP and route filtering policy. In it we first show a RIP configuration that distributes all known routes among three routers. Then we configure a route filtering policy using access lists to filter out advertisement of one network. The configuration example is based on the following reference diagram.

Figure 1: RIP configuration reference diagram



Basic RIP configuration

This example assumes that the router interfaces are already configured; the RIP configuration on each of the routers is shown below.

Table 1: Basic RIP configuration

Router	Step	Command(s)

R1	Display the configuration.	`vyatta@R1# show protocols`
		`rip {`
		`  network 10.0.40.0/24`
		`  redistribute {`
		`  connected {`
		`  }`
		`  }`
		`}`

R2	Display the configuration.	`vyatta@R2# show protocols`
		`rip {`
		`  network 10.0.40.0/24`
		`  network 10.0.50.0/24`
		`  redistribute {`
		`  connected {`
		`  }`
		`  }`
		`}`

Router	Step	Command(s)

R3	Display the configuration.	`vyatta@R2# show protocols`
		` rip {`
		`  network 10.0.50.0/24`
		`  redistribute {`
		`  connected {`
		`  }`
		`  }`
		` }`

Verifying the RIP configuration

The following operational mode commands can be used to verify the RIP configuration.

R3: show ip route

The following example shows the output of the `show ip route` command for router R3.

```
vyatta@R3:~$ show ip route

Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 10.0.20.0/24 [120/3] via 10.0.50.2, dp0p1p6, 00:20:16
R>* 10.0.30.0/24 [120/3] via 10.0.50.2, dp0p1p6, 00:34:04
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, dp0p1p6, 02:15:26
C>* 10.0.50.0/24 is directly connected, dp0p1p6
C>* 10.0.60.0/24 is directly connected, dp0p1p7
C>* 127.0.0.0/8 is directly connected, lo
vyatta@R3:~$
```

The output shows that routes to 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded out dp0p1p6 to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected.

R3: show ip rip

The `show ip rip` command for R3 displays similar information in a different format. This is shown in the following example.

```
vyatta@R3:~$ show ip rip

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
R(n) 10.0.20.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.30.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.40.0/24       10.0.50.2             2 10.0.50.2         0 00:23
C(i) 10.0.50.0/24       0.0.0.0               1 self              0
C(r) 10.0.60.0/24       0.0.0.0               1 self (connected:1)  0
vyatta@R3:~$
```

Again, the output shows that networks 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected.

Creating a route filtering policy

In this section, you configure a route filtering policy on R2 using access lists to deny incoming routes from 10.0.20.0/24.

Table 2: Route filtering configuration

Router	Step	Command(s)
R2	Create an access list and a rule to deny specified routes.	`vyatta@R2# set policy access-list 100 rule 10 action deny`
R2	Match any destination.	`vyatta@R2# set policy access-list 100 rule 10 destination any`
R2	Match source 10.0.20.0.	`vyatta@R2# set policy access-list 100 rule 10 source network 10.0.20.0`
R2	Specify the inverse mask for the network.	`vyatta@R2# set policy access-list 100 rule 10 source inverse-mask 0.0.0.255`
R2	Create a rule to permit all other routes.	`vyatta@R2# set policy access-list 100 rule 20 action permit`
R2	Match any destination.	`vyatta@R2# set policy access-list 100 rule 20 destination any`
R2	Match any source.	`vyatta@R2# set policy access-list 100 rule 20 source any`
R2	Commit the changes.	`vyatta@R2# commit`

Router	Step	Command(s)

R2	Display the configuration.	`vyatta@R2# show policy`
		` access-list 100 {`
		`  rule 10 {`
		`  action deny`
		`  destination {`
		`  any`
		`  }`
		`  source {`
		`  inverse-mask`
		` 0.0.0.255`
		`  network`
		` 10.0.20.0`
		`  }`
		`  }`
		`  rule 20 {`
		`  action permit`
		`  destination {`
		`  any`
		`  }`
		`  source {`
		`  any`
		`  }`
		`  }`
		` }`

Applying a route filtering policy

In this section, you apply the route filtering policy to incoming RIP advertisements on R2.

Table 3: Applying a route filtering policy

Router	Step	Command(s)

R2	Use the access list created in the previous example to filter incoming route advertisements.	`vyatta@R2# set protocols rip`
		` distribute-list access-list`
		` in 100`

R2	Commit the configuration.	`vyatta@R2# commit`

R2	Display the configuration.	`vyatta@R2# show protocols`
		` rip {`
		`  distribute-list {`
		`  access-list {`
		`  in 100`
		`  }`
		`  }`
		`  network 10.0.40.0/24`
		`  network 10.0.50.0/24`
		`  redistribute {`
		`  connected {`
		`  }`
		`  }`
		` }`

Verifying the route filtering policy configuration

The following operational mode commands can be used to verify the route filtering policy configuration.

R3: show ip route

The following example shows the output of the `show ip route` command for router R3.

```
vyatta@R3:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 10.0.30.0/24 [120/3] via 10.0.50.2, dp0p1p6, 00:45:21
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, dp0p1p6, 00:45:21
C>* 10.0.50.0/24 is directly connected, dp0p1p6
C>* 10.0.60.0/24 is directly connected, dp0p1p7
C>* 127.0.0.0/8 is directly connected, lo
vyatta@R3:~$
```

The output shows that routes to 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded out dp0p1p6 to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected. Notice that there is no route to 10.0.20.0/24 as it was filtered by the routing policy.

R3: show ip rip

The `show ip rip` command for R3 displays similar information in a different format. This is shown in the following example.

```
vyatta@R3:~$ show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
R(n) 10.0.30.0/24       10.0.50.2             3 10.0.50.2        0 00:22
R(n) 10.0.40.0/24       10.0.50.2             2 10.0.50.2        0 00:22
C(i) 10.0.50.0/24       0.0.0.0               1 self             0
C(i) 10.0.60.0/24       0.0.0.0               1 self             0
vyatta@R3:~$
```

Again, the output shows that networks 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected. Again, there is no route to 10.0.20.0/24.

Filtering inbound routes using prefix lists

This section presents the following topics:

- Prefix list configuration.
- Verifying the inbound filter.

Prefix list configuration

A common requirement for BGP configurations is to filter inbound routing announcements from a BGP peer. On the AT&T Vyatta vRouter, this is accomplished using routing policies that are then applied to the BGP process as “import” policies. In this instance we use prefix lists in conjunction with route maps to accomplish this.

The procedure below creates the following inbound filtering policies:

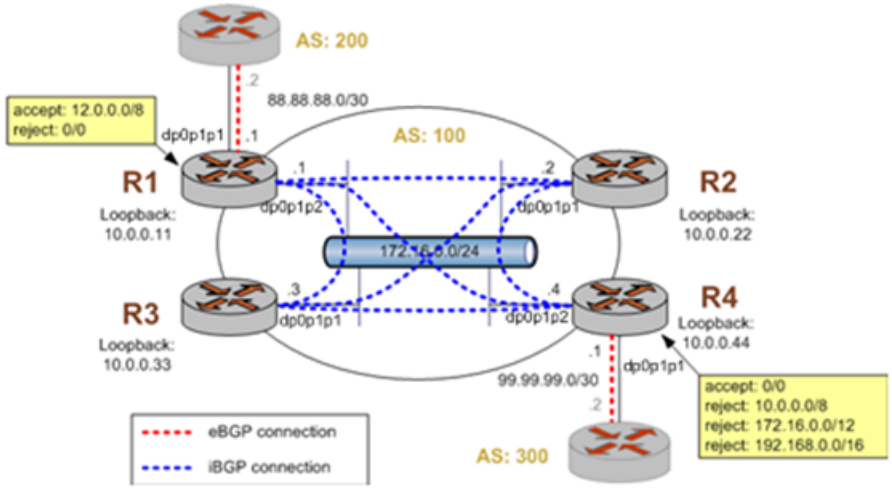
- R1 should only accept network 12.0.0.0/8 from its eBGP peer, and reject everything else.

- R4 should allow all Internet routes, but reject all RFC 1918 networks from its eBGP peer.

This import policy is shown in following figure.

Note: We assume that the routers in AS100 have been configured for iBGP and eBGP as shown and that the routers in AS200 and AS300 are configured appropriately as eBGP peers.

Figure 2: Filtering inbound routes



To create this inbound route filter, perform the following steps in configuration mode.

Table 4: Creating an import policy

Router	Step	Command(s)
R1	Create a list of prefixes to allow. In this case we just have one - 12.0.0.0/8.	`vyatta@R1# set policy route prefix-list ALLOW-PREFIXES rule 1 action permit` `vyatta@R1# set policy route prefix-list ALLOW-PREFIXES rule 1 prefix 12.0.0.0/8`
R1	Create a route map rule to permit all prefixes in our list.	`vyatta@R1# set policy route-map eBGP-IMPORT rule 10 action permit` `vyatta@R1# set policy route-map eBGP-IMPORT rule 10 match ip address prefix-list ALLOW-PREFIXES`
R1	Create a route map rule to deny all other prefixes.	`vyatta@R1# set policy route-map eBGP-IMPORT rule 20 action deny`
R1	Assign the route map policy created as the import route map policy for AS 200.	`vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 address-family ipv4-unicast route-map import eBGP-IMPORT`

Router	Step	Command(s)
R1	Commit the configuration.	`vyatta@R1# commit`
R1	Reset the BGP session to the peer so that the new policies are enabled.	`vyatta@R1# run reset ip bgp 88.88.88.2`

R1	Display the policy configuration.	`vyatta@R1# show policy`
		`route {`
		`prefix-list ALLOW-PREFIXES {`
		`rule 1 {`
		`action permit`
		`prefix 12.0.0.0/8`
		`}`
		`}`
		`route-map eBGP-IMPORT {`
		`rule 10 {`
		`action permit`
		`match {`
		`ip {`
		`address {`
		`prefix-list ALLOW-PREFIXES`
		`}`
		`}`
		`}`
		`}`
		`rule 20 {`
		`action deny`
		`}`
		`}`
		`vyatta@R1#`

R1	Display the BGP configuration for eBGP neighbor 88.88.88.2.	`vyatta@R1# show protocols`
		`bgp 100 neighbor 88.88.88.2{`
		`address-family {`
		`ipv4-unicast {`
		`route-map {`
		`import eBGP-IMPORT`
		`}`
		`}`
		`ipv6-unicast {`
		`}`
		`}`
		`ebgp-multihop 1`
		`remote-as 200`
		`}`
		`vyatta@R1#`

Router	Step	Command(s)

R4	Create a rule to match any prefix from 10.0.0.0/8 to 32.	`vyatta@R4# set policy route prefix-list RFC1918PREFIXES rule 1 action permit`
		`vyatta@R4# set policy route prefix-list RFC1918PREFIXES rule 1 le 32`
		`vyatta@R4# set policy route prefix-list RFC1918PREFIXES rule 1 prefix 10.0.0.0/8`

R4	Commit the configuration.	`vyatta@R4# commit`
R4	Create a route map rule to deny all prefixes in our list.	`vyatta@R4# set policy route-map eBGP-IMPORT rule 10 action deny` `vyatta@R4# set policy route-map eBGP-IMPORT rule 10 match ip address prefix-list RFC1918PREFIXES`
R4	Create a route map rule to permit all other prefixes.	`vyatta@R4# set policy route-map eBGP-IMPORT rule 20 action permit`
R4	Commit the configuration.	`vyatta@R4# commit`
R4	Assign the route map policy created as the import route map policy for AS 300.	`vyatta@R4# set protocols bgp 100 neighbor 99.99.99.2 route-map import eBGP-IMPORT`
R4	Commit the configuration.	`vyatta@R4# commit`
R4	Reset the BGP session to the peer so that the new policies are enabled.	`vyatta@R4# run reset ip bgp 99.99.99.2`

Router	Step	Command(s)

R4	Display the policy configuration.	vyatta@R4# show policy
		route {
		prefix-list
		RFC1918PREFIXES {
		rule 1 {
		action permit
		le 32
		prefix
		10.0.0.0/8
		}
		}
		route-map eBGP-IMPORT {
		rule 10 {
		action deny
		match {
		ip {
		address
		{
		prefix-list RFC1918PREFIXES
		}
		}
		}
		}
		rule 20 {
		action permit
		}
		}
		}
		vyatta@R4#

R4	Display the BGP configuration for eBGP neighbor 99.99.99.2.	vyatta@R4# show protocols
		bgp 100 neighbor 99.99.99.2
		address-family {
		ipv4-unicast {
		route-map {
		import
		eBGP-IMPORT
		}
		}
		ipv6-unicast {
		}
		}
		ebgp-multihop 1
		remote-as 300
		}
		vyatta@R4#

Verifying the inbound filter

The following commands can be used to verify the inbound filter configuration.

R1: show ip bgp before applying import filter

The following example shows R1's BGP table before the import filter is applied.

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       88.88.88.2               0             0 200 i
*> 2.1.0.0/24       88.88.88.2               0             0 200 i
*> 2.2.0.0/24       88.88.88.2               0             0 200 i
*>i3.0.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      99.99.99.2               0    100      0 300 i
*> 88.88.88.0/30    88.88.88.2               0             0 200 i
*>i99.99.99.0/30    99.99.99.2               0    100      0 300 i
*> 172.16.0.0/24    0.0.0.0                  1         32768 i
* i                 10.0.0.44                1    100      0 i
*>i172.16.128.0/24  99.99.99.2               0    100      0 300 i
*>i192.168.2.0      99.99.99.2               0    100      0 300 i

Total number of prefixes 13
vyatta@R1:~$
```

R1: show ip bgp after applying import filter

The following example shows R1's BGP table after the import filter is applied. Note that only 12.0.0.0 from 88.88.88.2 is still in the table.

```
vyatta@R1:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i3.0.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.1.0.0/24       99.99.99.2               0    100      0 300 i
*>i3.2.0.0/24       99.99.99.2               0    100      0 300 i
*> 12.0.0.0         88.88.88.2               0             0 200 i
*>i13.0.0.0/24      99.99.99.2               0    100      0 300 i
*>i99.99.99.0/30    99.99.99.2               0    100      0 300 i
*> 172.16.0.0/24    0.0.0.0                  1         32768 i
* i                 10.0.0.44                1    100      0 i
*>i172.16.128.0/24  99.99.99.2               0    100      0 300 i
*>i192.168.2.0      99.99.99.2               0    100      0 300 i

Total number of prefixes 9
vyatta@R1:~$
```

R4: show ip bgp before applying import filter

The following example shows R4's BGP table before the import filter is applied.

```
vyatta@R4:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3.0.0.0/24       99.99.99.2               0             0 300 i
*> 3.1.0.0/24       99.99.99.2               0             0 300 i
*> 3.2.0.0/24       99.99.99.2               0             0 300 i
*>i12.0.0.0         88.88.88.2               0    100      0 200 i
*> 13.0.0.0/24      99.99.99.2               0             0 300 i
```

```
*> 99.99.99.0/30    99.99.99.2               0             0 300 i
* i172.16.0.0/24    10.0.0.11                1    100      0 i
*>                  0.0.0.0                  1         32768 i
*> 172.16.128.0/24  99.99.99.2               0             0 300 i
*> 192.168.2.0      99.99.99.2               0             0 300 i

Total number of prefixes 9
vyatta@R4:~$
```

R4: show ip bgp after applying import filter

The output below shows R4's BGP table after the import filter is applied.

```
vyatta@R4:~$ show ip bgp
BGP table version is 2, local router ID is 10.0.0.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, l - labeled
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i12.0.0.0         88.88.88.2               0    100      0 200 i

Total number of prefixes 1
```

Filtering outbound routes using AS path lists

This section presents the following topics:

- As-path-list configuration
- Verifying the outbound filter

As-path-list configuration

Filtering outbound prefixes is another common BGP configuration requirement. On the AT&T Vyatta vRouter, this is accomplished using routing policies that are then applied to the BGP process as export policies.

The example in this section assumes that AS100 does not want to be a transit AS for AS200 or AS300. This means that:

- eBGP routes from R1's eBGP peer (AS200) should not be sent to R4's eBGP peer.
- Routes from R4's eBGP peer (AS300) should not be sent to R1's eBGP peer.

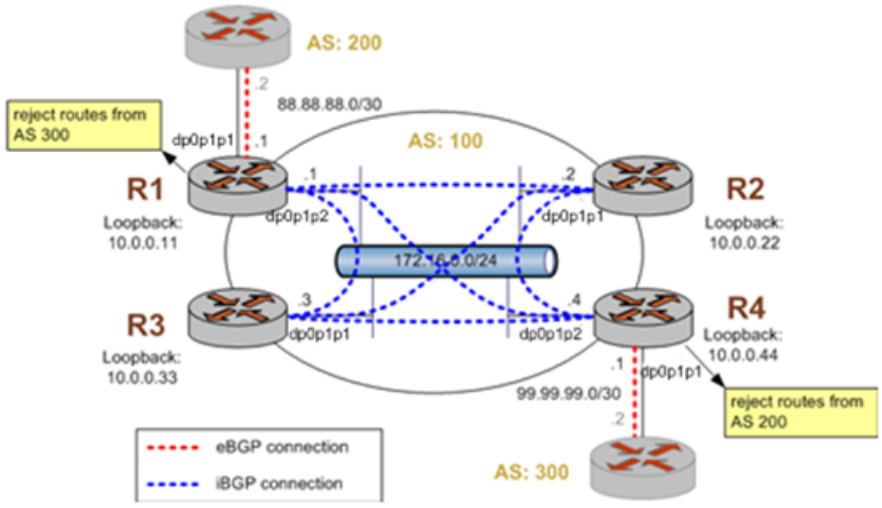
If we did not implement this filtering, AS300 might send traffic destined for AS200 to router R4, and this traffic would then be carried across the AS100 network.

There are several ways that this routing policy could be implemented: two most common are basing the filter on the network prefix or basing it on the AS Path. In this example, we update the existing BGP export policy to add some additional restrictions that will prevent AS100 from acting as a transit network for AS200 and AS300.

This export policy is shown in the following figure.

> **Note:** We assume that the routers in AS100 have been configured for iBGP and eBGP as shown and that the routers in AS200 and AS300 are configured appropriately as eBGP peers.

Figure 3: Filtering outbound routes



To create this export policy, perform the following steps in configuration mode.

Table 5: Creating an export policy

Router	Step	Command(s)

R1	Create a list of AS paths to deny. In this case we just have one - AS300.	`vyatta@R1# set policy route as-path-list AS300 rule 1 action permit`
		
		`vyatta@R1# set policy route as-path-list AS300 rule 1 regex 300`

R1	Create a route map rule to deny all AS paths in our list.	`vyatta@R1# set policy route route-map eBGP-EXPORT rule 10 action deny`
		
		`vyatta@R1# set policy route route-map eBGP-EXPORT rule 10 match as-path AS300`

R1	Create a route map rule to permit all other prefixes.	`vyatta@R1# set policy route route-map eBGP-EXPORT rule 20 action permit`

Router	Step	Command(s)

R1	Assign the route map policy created as the export and import route map policy for AS 200.	`vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 remote-as 200`
		`vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 address-family ipv4-unicast route-map export eBGP-EXPORT`
		`vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 address-family ipv4-unicast route-map import eBGP-IMPORT`
		`vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 ebgp-multihop 1`

R1	Commit the configuration.	`vyatta@R1# commit`
R1	Reset the BGP session to the peer so that the new policies are enabled.	`vyatta@R1# run reset ip bgp 88.88.88.2`

R1	Display the policy configurations.	`vyatta@R1# show policy`
		`route {`
		`  as-path-list AS300 {`
		`  rule 1 {`
		`  action permit`
		`  regex 300`
		`  }`
		`  }`
		`  route-map eBGP-EXPORT {`
		`  rule 10 {`
		`  action deny`
		`  match {`
		`  as-path AS300`
		`  }`
		`  }`
		`  rule 20 {`
		`  action permit`
		`  }`
		`  }`

Router	Step	Command(s)

R1	Display the BGP configuration for eBGP neighbor 88.88.88.2.	`vyatta@R1# show protocols`
		` bgp 100 neighbor 88.88.88.2`
		` address-family {`
		`  ipv4-unicast {`
		`  route-map {`
		`  export eBGP-EXPORT`
		`  import eBGP-IMPORT`
		`  }`
		`}`
		`}`
		` ebgp-multihop 1`
		` remote-as 200`

R4	Create a list of AS paths to deny. In this case we just have one - AS200.	`vyatta@R4# set policy route`
		` route-map eBGP-EXPORT rule`
		` 20 action permit`
		`vyatta@R4# set policy route`
		` as-path-list AS200 rule 1`
		` regex 200`
		`vyatta@R4# commit`

R4	Create a route map rule to deny all AS paths in our list.	`vyatta@R4# set policy route`
		` route-map eBGP-EXPORT rule`
		` 10 action deny`
		`vyatta@R4# set policy route`
		` route-map eBGP-EXPORT rule`
		` 10 match as-path AS200`

R4	Create a route map rule to permit all other prefixes.	`vyatta@R4# set policy route`
		` route-map eBGP-EXPORT rule`
		` 20 action permit`
		`vyatta@R4# commit`

R4	Assign the route map policy created as the export route map policy for AS300.	`vyatta@R4#set protocol`
		` bgp 100 neigh 99.99.99.2`
		` address-family ipv4-unicast`
		`  route-map export eBGP-EXPORT`

R4	Commit the configuration.	`vyatta@R4# commit`
R4	Reset the BGP session to the peer so that the new policies are enabled.	`vyatta@R4# run reset ip bgp` ` 99.99.99.2`

Router	Step	Command(s)

R4	Display the policy configurations.	vyatta@R4# show policy
		route {
		as-path-list AS200 {
		rule 1 {
		action permit
		regex 200
		}
		}
		prefix-list
		RFC1918PREFIXES {
		rule 1 {
		action permit
		le 32
		prefix
		10.0.0.0/8
		}
		}
		route-map eBGP-EXPORT {
		rule 10 {
		action deny
		match {
		as-path
		AS200
		}
		}
		rule 20 {
		action permit
		}
		}
		route-map eBGP-IMPORT {
		rule 10 {
		action deny
		match {
		ip {
		address
		{
		
		prefix-list RFC1918PREFIXES
		}
		}
		}
		}
		rule 20 {
		action permit
		}
		}
		}

Router	Step	Command(s)

R4	Display the BGP configuration for eBGP neighbor 99.99.99.2.	`vyatta@R4# show protocols`
		` bgp 100 neighbor 99.99.99.2`
		` address-family {`
		`  ipv4-unicast {`
		`  route-map {`
		`  import eBGP-IMPORT`
		`  }`
		`  soft-reconfiguration {`
		`  inbound`
		`  }`
		`  }`
		`  ipv6-unicast {`
		`  }`
		` }`
		` ebgp-multihop 1`
		` remote-as 300`

Verifying the outbound filter

The following commands can be used to verify the outbound filter configuration.

AS 200: show ip bgp before applying export filter

The following example shows AS 200's BGP table before the export filter is applied.

```
vyatta@AS200:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       0.0.0.0                  0         32768 i
*> 2.1.0.0/24       0.0.0.0                  0         32768 i
*> 2.2.0.0/24       0.0.0.0                  0         32768 i
*> 3.0.0.0/24       88.88.88.1                             0 100 300 i
*> 3.1.0.0/24       88.88.88.1                             0 100 300 i
*> 3.2.0.0/24       88.88.88.1                             0 100 300 i
*> 12.0.0.0         0.0.0.0                  0         32768 i
*> 13.0.0.0/24      88.88.88.1                             0 100 300 i
*> 88.88.88.0/30    0.0.0.0                  0         32768 i
*> 99.99.99.0/30    88.88.88.1                             0 100 300 i
*> 172.16.0.0/24    88.88.88.1               1             0 100 i

Total number of prefixes 11
vyatta@AS200:~$
```

AS 200: show ip bgp after applying export filter

The following example shows AS 200's BGP table after the export filter is applied.

```
vyatta@AS200:~$ show ip bgp
BGP table version is 0, local router ID is 10.0.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2.0.0.0/24       0.0.0.0                  0         32768 i
```

```
*> 2.1.0.0/24       0.0.0.0                  0         32768 i
*> 2.2.0.0/24       0.0.0.0                  0         32768 i
*> 12.0.0.0         0.0.0.0                  0         32768 i
*> 88.88.88.0/30    0.0.0.0                  0         32768 i
*> 172.16.0.0/24    88.88.88.1               1             0 100 i

Total number of prefixes 6
vyatta@AS200:~$
```

Routing Policy Commands

policy route access-list <list-num>

Defines an access list.

Syntax:
`set policy route access-list` *list-num*

Syntax:
`delete policy route access-list` *list-num*

Syntax:
`show policy route access-list` *list-num*

list-num
Multi-node. A numeric identifier for the access list. Access list numbers can take the following values:

1 to 99: IP standard access list

100 to 199: IP extended access list

1300 to 1999: IP standard access list (expanded range)

2000 to 2699: IP extended access list (expanded range)

You can create multiple access lists by creating multiple **`policy access-list`** configuration nodes.

Configuration mode

```
policy {
    route {
        access-list list-num {}
    }
}
```

Use the `set` form of this command to create an access list.

Use the `delete` form of this command to remove an access list.

Use the `show` form of this command to display access list configuration.

policy route access-list <list-num> description <desc>

Allows you to specify a brief d escription for an access list.

Syntax:
`set policy route access-list` *list-num* **`description`** *desc*

Syntax:
`delete policy route access-list` *list-num* **`description`**

Syntax:
`show policy route access-list` *list-num* **`description`**

list-num
The number of a defined access list.

desc
A brief text description for the access list.

Configuration mode

```
policy {
    route {
        access-list list-num {
            description desc
        }
    }
}
```

Use the `set` form of this command to create a description for an access list.

Use the `delete` form of this command to remove an access list description.

Use the `show` form of this command to display the description for an access list.

policy route access-list <list-num> rule <rule-num>

Creates a rule for an access list.

Syntax:
`set policy route access-list` *list-num* **`rule`** *rule-num*

Syntax:
`delete policy route access-list` *list-num* **`rule`** *rule-num*

Syntax:
`show policy route access-list` *list-num* **`rule`** *rule-num*

list-num
The number of a defined access list.

rule-num
Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route {
        access-list list-num {
            rule rule-num {}
        }
    }
}
```

Use the `set` form of this command to create an access list rule.

Use the `delete` form of this command to remove an access list rule.

Use the `show` form of this command to display configuration settings for an access list rule.

policy route access-list <list-num> rule <rule-num> action

Specifies the action to be taken for packets matching an access list rule.

Syntax:
`set policy route access-list` *list-num* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:
`delete policy route access-list` *list-num* **`rule`** *rule-num* **`action`**

Syntax:
`show policy route access-list` *list-num* **`rule`** *rule-num* **`action`**

Packets matching this rule are forwarded.

list-num
The number of a defined access list.

rule-num
The number of a defined access list rule.

`deny`
Packets matching this rule are silently dropped.

`permit`
Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route {
        access-list list-num {
            rule rule-num {
                action {
                    deny
                    permit
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **`deny`**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **`permit`**, packets meeting the match criteria of the rule are forwarded.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route access-list <list-num> rule <rule-num> description <desc>

Allows you to specify a brief description for an access list rule.

Syntax:
`set policy route access-list` *list-num* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route access-list` *list-num* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route access-list` *list-num* **`rule`** *rule-num* **`description`**

list-num
The number of a defined access list.

rule-num
The number of a defined access list rule.

desc
A brief text description for the access list rule.

Configuration mode

```
policy {
    route {
    access-list list-num {
        rule rule-num {
            description desc
            }
        }
    }
}
```

Use the `set` form of this command to create a description for an access list rule.

Use the `delete` form of this command to remove an access list rule description.

Use the `show` form of this command to display an access list rule description.

policy route access-list <list-num> rule <rule-num> destination

Defines match criteria for an access list rule based on destination.

Syntax:

`set policy route access-list` *list-num* **`rule`** *rule-num* **`destination`** { **`any`** | **`host`** *ipv4* | **`inverse-mask`** *ipv4* | **`network`** *ipv4net* }

Syntax:

`delete policy route access-list` *list-num* **`rule`** *rule-num* **`destination`**

Syntax:

`show policy route access-list` *list-num* **`rule`** *rule-num* **`destination`**

list-num
: The number of a defined access list.

rule-num
: The number of a defined access list.

`any`
: Match packets destined for any destination. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`host` ***ipv4***
: Match packets destined for the specified IPv4 host. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`inverse-mask` ***ipv4***
: Match packets destined for the network specified by the mask. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`network` ***ipv4net***
: Match packets destined for the specified network. The format is *ip-address/prefix*. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

Configuration mode

```
policy {
    route {
    access-list list-num {
        rule rule-num {
            destination {
                any
                host ipv4
                inverse-mask ipv4
                network ipv4net
```

```
                }
            }
        }
    }
}
```

Use the `set` form of this command to specify the destination match criteria for this access list rule.

Use the `delete` form of this command to remove configured destination match criteria for this rule. If no match criteria are specified, no packet filtering on destination will take place; that is, packets to all destinations are permitted.

Use the `show` form of this command to display configuration settings for access list rule destination packet filtering.

policy route access-list <list-num> rule <rule-num> source

Defines match criteria for an access list rule based on source.

Syntax:

`set policy route access-list` *list-num* **`rule`** *rule-num* **`source`** { **`any`** | **`host`** *ipv4* | **`inverse-mask`** *ipv4* | **`network`** *ipv4net* }

Syntax:

`delete policy route access-list` *list-num* **`rule`** *rule-num* **`source`**

Syntax:

`show policy route access-list` *list-num* **`rule`** *rule-num* **`source`**

list-num
: The number of a defined access list.

rule-num
: The number of a defined access list rule.

`any`
: Match packets coming from any source. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`host` ***ipv4***
: Match packets coming from the specified IPv4 host. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`inverse-mask` ***ipv4***
: Match packets coming from the network specified by the mask. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

`network` ***ipv4net***
: Match packets coming from the specified network. The format is *ip-address/prefix*. Exactly one of **`any`**, **`host`**, **`inverse-mask`**, and **`network`** is mandatory.

Configuration mode

```
policy {
    route {
    access-list list-num {
        rule rule-num {
            source {
                any
                host ipv4
                inverse-mask ipv4
                network ipv4net
                }
            }
        }
    }
```

```
}
```

Use the `set` form of this command to specify the source match criteria for this access list rule.

Use the `delete` form of this command to remove the configured source match criteria for this rule. If no match criteria are specified, no packet filtering on source will take place; that is, packets from all sources are permitted.

Use the `show` form of this command to display configuration settings for access list rule source packet filtering.

policy route access-list6 <list-name>

Defines an IPv6 access list.

Syntax:
`set policy route access-list6` *list-name*

Syntax:
`delete policy route access-list6` *list-name*

Syntax:
`show policy route access-list6` *list-name*

list-name
Multi-node. The name of an IPv6 access list.

You can create multiple access lists by creating multiple **`policy access-list`** configuration nodes.

Configuration mode

```
policy {
    route {
    access-list6 list-name {}
    }
}
```

Use the `set` form of this command to create an access list.

Use the `delete` form of this command to remove an access list.

Use the `show` form of this command to display access list configuration.

policy route access-list6 <list-name> description <desc>

Allows you to specify a brief description for an IPv6 access list.

Syntax:
`set policy route access-list6` *list-name* **`description`** *desc*

Syntax:
`delete policy route access-list6` *list-name* **`description`**

Syntax:
`show policy route access-list6` *list-name* **`description`**

list-name
The name of an IPv6 access list.

desc
A brief text description for the access list.

Configuration mode

```
policy {
```

```
    route{
    access-list6 list-name {
        description desc
        }
    }
}
```

Use the `set` form of this command to create a description for an access list.

Use the `delete` form of this command to remove an access list description.

Use the `show` form of this command to display the description for an access list.

policy route access-list6 <list-name> rule <rule-num>

Creates a rule for an IPv6 access list.

Syntax:

`set policy route access-list6` *list-name* **`rule`** *rule-num*

Syntax:

`delete policy route access-list6` *list-name* **`rule`** *rule-num*

Syntax:

`show policy route access-list6` *list-name* **`rule`** *rule-num*

list-name

The name of an IPv6 access list.

rule-num

Multi-node. A numeric identifier for the rule. The range is 1 to 65535.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route {
    access-list6 list-name {
        rule rule-num {}
        }
    }
}
```

Use the `set` form of this command to create an access list rule.

Use the `delete` form of this command to remove an access list rule.

Use the `show` form of this command to display configuration settings for an access list rule.

policy route access-list6 <list-name> rule <rule-num> action

Specifies the action to be taken for packets matching an IPv6 access list rule.

Syntax:

`set policy route access-list6` *list-name* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:

`delete policy route access-list6` *list-name* **`rule`** *rule-num* **`action`**

Syntax:

`show policy route access-list6` *list-name* **`rule`** *rule-num* **`action`**

Packets matching this rule are forwarded.

list-name
The name of an IPv6 access list.

rule-num
The number of a defined access list rule.

deny
Packets matching this rule are silently dropped.

permit
Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route {
    access-list6 list-name {
        rule rule-num {
            action {
                deny
                permit
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **deny**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **permit**, packets meeting the match criteria of the rule are forwarded.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route access-list6 <list-name> rule <rule-num> description <desc>

Allows you to specify a brief description for an IPv6 access list rule.

Syntax:
`set policy route access-list6` *list-name* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route access-list6` *list-name* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route access-list6` *list-name* **`rule`** *rule-num* **`description`**

list-name
The name of an IPv6 access list.

rule-num
The number of a defined access list rule.

desc
A brief text description for the access list rule.

Configuration mode

```
policy {
    route {
```

```
    access-list6 list-name {
        rule rule-num {
            description desc
            }
        }
    }
}
```

Use the `set` form of this command to create a description for an access list rule.

Use the `delete` form of this command to remove an access list rule description.

Use the `show` form of this command to display an access list rule description.

policy route access-list6 <list-name> rule <rule-num>

Allows you to specify the list name and rule number for an IPv6 access list rule.

Syntax:
`set policy route access-list6` *list-name* **`rule`** *rule-num*

Syntax:
`delete policy route access-list6` *list-name* **`rule`**

Syntax:
`show policy route access-list6` *list-name* **`rule`**

list-name
The name of an IPv6 access list.

rule-num
The number of a defined IPv6 access list.

Configuration mode

```
policy {
    route {
    access-list6 list-name {
        rule rule-num {}
        }
    }
}
```

Use the `set` form of this command to specify the access list rule name and number.

Use the `delete` form of this command to remove the rule.

Use the `show` form of this command to display the access list rule name and number.

policy route access-list6 <list-name> rule <rule-num> source

Defines match criteria for an IPv6 access list rule based on source.

Syntax:
`set policy route access-list6` *list-name* **`rule`** *rule-num* **`source`** { **`any`** | **`exact-match`** | **`network`** *ipv6net* }

Syntax:
`delete policy route access-list6` *list-name* **`rule`** *rule-num* **`source`**

Syntax:
`show policy route access-list6` *list-name* **`rule`** *rule-num* **`source`**

list-name

The name of an IPv6 access list.

rule-num

The number of a defined IPv6 access list rule.

any

Match packets coming from any source. Exactly one of **any**, **exact-match**, and **network** is mandatory.

exact-match

Match packets coming from one of the network prefixes. Exactly one of **any**, **exact-match**, and **network** is mandatory.

network *ipv6net*

Match packets coming from the specified network. The format is *iv6p-address/prefix*. Exactly one of **any**, **exact-match**, and **network** is mandatory.

Configuration mode

```
policy {
    route {
    access-list6 list-name {
        rule rule-num {
            source {
                any
                exact-match
                network ipv6net
                }
            }
        }
    }
}
```

Use the `set` form of this command to specify the source match criteria for this access list rule.

Use the `delete` form of this command to remove the configured source match criteria for this rule. If no match criteria are specified, no packet filtering on source will take place; that is, packets from all sources are permitted.

Use the `show` form of this command to display configuration settings for access list rule source packet filtering.

policy route as-path-list <list-name>

Defines an autonomous system (AS) path list.

Syntax:

`set policy route as-path-list` *list-name*

Syntax:

`delete policy route as-path-list` *list-name*

Syntax:

`show policy route as-path-list` *list-name*

list-name

Multi-node. A text identifier for the AS path list.

You can create multiple AS path lists by creating multiple **policy as-path-list** configuration nodes.

Configuration mode

```
policy {
    route {
    as-path-list list-name {}
    }
}
```

Use the `set` form of this command to define an autonomous system (AS) path list for use in policy-based routing.

Use the `delete` form of this command to remove an AS path list.

Use the `show` form of this command to display AS path list configuration.

policy route as-path-list <list-name> description <desc>

Allows you to specify a brief description for an AS path list.

Syntax:
`set policy route as-path-list` *list-name* **`description`** *desc*

Syntax:
`delete policy route as-path-list` *list-name* **`description`**

Syntax:
`show policy route as-path-list` *list-name* **`description`**

list-name
The name of a defined AS path list.

desc
A brief text description for the AS path list.

Configuration mode

```
policy {
    route {
    as-path-list list-name {
        description desc
        }
    }
}
```

Use the `set` form of this command to specify a description for an AS path list.

Use the `delete` form of this command to remove an AS path list description.

Use the `show` form of this command to display an AS path list description.

policy route as-path-list <list-name> rule <rule-num>

Creates a rule for an AS path list.

Syntax:
`set policy route as-path-list` *list-name* **`rule`** *rule-num*

Syntax:
`delete policy route as-path-list` *list-name* **`rule`** *rule-num*

Syntax:
`show policy route as-path-list` *list-name* **`rule`** *rule-num*

list-name
The name of a defined AS path list.

rule-num
Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route {
    as-path-list list-name {
        rule rule-num {}
        }
    }
}
```

Use the `set` form of this command to create an AS path list rule.

Use the `delete` form of this command to remove an AS path list rule.

Use the `show` form of this command to display configuration settings for an AS path list rule.

policy route as-path-list <list-name> rule <rule-num> action

Specifies the action to be taken for packets matching an AS path list rule.

Syntax:

`set policy route as-path-list` *list-name* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:

`delete policy route as-path-list` *list-name* **`rule`** *rule-num* **`action`**

Syntax:

`show policy route as-path-list` *list-name* **`rule`** *rule-num* **`action`**

Packets matching this rule are forwarded.

list-name
: The name of a defined AS path list.

rule-num
: The number of a defined AS path list rule.

`deny`
: Packets matching this rule are silently dropped.

`permit`
: Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route {
    as-path-list list-name {
        rule rule-num {
            action {
                deny
                permit
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **`deny`**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **`permit`**, destination-based routing is performed; that is, packets are sent using the normal forwarding channels.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route as-path-list <list-name> rule <rule-num> description <desc>

Allows you to specify a brief description for an AS path list rule.

Syntax:
`set policy route as-path-list` *list-name* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route as-path-list` *list-name* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route as-path-list` *list-name* **`rule`** *rule-num* **`description`**

list-name
The name of a defined AS path list.

rule-num
The number of a defined AS path list rule.

desc
A brief text description for the AS path list rule.

Configuration mode

```
policy {
    route {
    as-path-list list-name {
        rule rule-num {
            description desc
            }
        }
    }
}
```

Use the `set` form of this command to specify a description for an AS path list.

Use the `delete` form of this command to remove an AS path list description.

Use the `show` form of this command to display an AS path list description.

policy route as-path-list <list-name> rule <rule-num> regex <regex>

Defines match criteria for an AS path list rule based on a regular expression.

Syntax:
`set policy route as-path-list` *list-name* **`rule`** *rule-num* **`regex`** *regex*

Syntax:
`delete policy route as-path-list` *list-name* **`rule`** *rule-num* **`regex`**

Syntax:
`show policy route as-path-list` *list-name* **`rule`** *rule-num* **`regex`**

If no regular expression is defined, all packets are considered to match the rule.

list-name
The name of a defined AS path list.

rule-num
The number of a defined AS path list rule.

regex
A POSIX-style regular expression representing an AS path list.

Configuration mode

```
policy {
    route {
    as-path-list list-name {
        rule rule-num {
            regex regex
            }
        }
    }
}
```

Use the `set` form of this command to define the match criteria to be used to determine forwarding policy based on AS paths.

Packets are matched based on whether the AS paths listed in the packet match the regular expression defined using this command. Depending on the action defined for the rule using policy route as-path-list <list-name> rule <rule-num> action *(page 39)*, matched packets are either permitted or denied.

Use the `delete` form of this command to remove the regular expression entry. If no regular expression is defined, all packets are considered to match the rule.

Use the `show` form of this command to display the regular expression entry.

policy route community-list [standard | expanded] { <list-num> | <list-name> }

Creates a standard BGP community list.

Syntax:
`set policy route community-list [ standard | expanded ] { list-num | list-name }`

Syntax:
`delete policy route community-list [ standard | expanded ] { list-num | list-name }`

Syntax:
`show policy route community-list [ standard | expanded ] { list-num | list-name }`

list-num
Multinode. A numeric identifier for the standard BGP community list.

A standard community lists number ranges from 1 through 99 and list name and an expanded community list ranges from 100 through 199.

list-name
A string identifier for the community list.

The string is a set of characters.

Configuration mode

```
policy {
    route {
    community-list {
        standard [list-num | list-name ]
        expanded [list-num | list-name ]
        }
    }
```

```
}
```

Use the `set` form of this command to create a standard BGP community list.

Note: You can create multiple community lists by creating multiple policy community-list configuration nodes.

Use the `delete` form of this command to delete a standard BGP community list.

Use the `show` form of this command to display standard BGP community list.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route community-list [standard | expanded] { <list-num> | <list-name> } description <desc>

Provides a brief description of a standard community list.

Syntax:

```
set policy route community-list [ standard | expanded ] { list-num | list-name } description desc
```

Syntax:

```
delete policy route community-list [ standard | expanded ] { list-num | list-name } description
```

Syntax:

```
show policy route community-list [ standard | expanded ] { list-num | list-name } description
```

list-num

The number of a defined community list.

A standard community lists number ranges from 1 through 99 and list name and an expanded community list ranges from 100 through 199.

list-name

A name, which is a character string, identifier for the community list.

The string is a set of characters.

desc

A brief text description of the community list.

Configuration mode

```
policy {
    route {
        community-list {
            standard [list-num | list-name]
            expanded [list-num | list-name]
                {
                    description desc
                }
            }
        }
}
```

Use the `set` form of this command to provide a brief description of a community list.

Use the `delete` form of this command to delete the description of a community list.

Use the `show` form of this command to display the description of a community list.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route community-list [standard | expanded] { <list-num> | <list-name> } rule <rule-num>

Creates a rule for a community list.

Syntax:
`set policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num`

Syntax:
`delete policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num`

Syntax:
`show policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num`

list-num
The number of a defined community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name
A string identifier for the community list.

The string is a set of characters.

rule-num
Multinode. A numeric identifier for the rule that is being created. The identifier ranges from 1 through 4294967295.

You can define multiple rules by creating multiple `rule` configuration nodes.

Configuration mode

```
policy {
    route {
    community-list {
            standard [list-num | list-name ]
            expanded [list-num | list-name ]
                {
                   rule rule-num
            }
        }
    }
}
```

Use the `set` form of this command to create a rule for community list.

Use the `delete` form of this command to delete a rule for community list.

Use the `show` form of this command to display the configuration of a rule for a community list.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route community-list standard { <list-num> | <list-name> } rule <rule-num> community <community>

Creates multiple rules for a single community list with different community values.

Syntax:

```
set policy route community-list standard { list-num | list-name } rule rule-num1 community { AA:NN |
local-AS | no-advertise | no-export | internet | none }
```

Syntax:

```
set policy route community-list standard { list-num | list-name } rule rule-num2 community { AA:NN |
local-AS | no-advertise | no-export | internet | none }
```

list-num

The number of a defined community list.

A standard community lists number ranges from 1 through 99 and list name and an expanded community list ranges from 100 through 199.

list-name

A name, which is a character string, for the community list.

The string is a set of characters.

rule-num

Multinode. A numeric identifier for the rule that is being created. The identifier ranges from 1 through 4294967295.

You can define multiple rules by creating multiple **rule** configuration nodes.

AA:NN

A community in 4-octet, AS-value format.

local-AS

Advertises communities in local AS only. (NO_EXPORT_SUBCONFED).

no-advertise

Does not advertise this route to any peer (NO_ADVERTISE).

no-export

Does not advertise outside of this AS of confederation boundary (NO_EXPORT).

internet

Specifies the 0 symbolic Internet community.

none

Specifies no communities.

```
policy {
    route {
    community-list {
              standard [list-num | list-name ]
              {
                 rule rule-num1 {
                        AA:NN
                        local-AS
                        no-advertise
                        no-export
                        internet
                        none
                    rule rule-num2 {
                        AA:NN
                        local-AS
                        no-advertise
                        no-export
                        internet
                        none
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to create a rule for a community list.

Use the `delete` form of this command to delete a rule for a community list.

Use the `show` form of this command to display a rule for a community list.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route community-list [standard | expanded] { <list-num> | <list-name> } rule <rule-num> action

Specifies the action to take when packets match a community list rule.

Syntax:

```
set policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num action { deny | permit }
```

Syntax:

```
delete policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num action
```

Syntax:

```
show policy route community-list [ standard | expanded ] { list-num | list-name } rule rule-num action
```

Packets that match this rule are forwarded.

list-num

The number of a defined community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name

A string identifier for the community list.

The string is a set of characters.

rule-num

The rule number for a defined community-list.

deny

Silently drops the packet that match this rule.

permit

Forwards packets that match this rule.

Configuration mode

```
policy {
    route {
        community-list {
                standard [list-num | list-name ]
                expanded [list-num | list-name ]
                    {
                       rule rule-num {
                          action {
                             deny
                             permit
                             }
                          }
                       }
                }
    }
}
```

Use the `set` form of this command to specify the action to take when packets match a community list rule.

If the action for a rule is **`deny`**, packets that meet the match criteria of the rule are silently dropped. If the action for the rule is **`permit`**, destination-based routing is performed; that is, packets are sent by using the normal forwarding channels.

Use the `delete` form of this command to restore the default action to take for packets that match a community list rule.

Use the `show` form of this command to display the action settings to take when packets match a community list rule.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route community-list expanded { <list-num> | <list-name> } rule <rule-num> regex <regex>

Configures a standard community list to define the match criteria for a community list rule, which is based on a regular expression for a community list.

Syntax:

```
set policy route community-list expanded { list-num | list-name } rule rule-num regex regex
```

Syntax:

```
delete policy route community-list expanded { list-num | list-name } rule rule-num regex
```

Syntax:

```
show policy route community-list expanded { list-num | list-name } rule rule-num regex
```

If no regular expression is defined, all packets are considered to match the rule.

list-num

The number of a defined extended community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name

A string identifier for the extended community list.

The string is a set of characters.

rule-num

The number of a defined community list rule.

regex

A POSIX-style regular expression that represents a BGP community list.

Configuration mode

```
policy {
    route {
        community-list {
               expanded [list-num | list-name ] {
                     rule rule-num {
                        regex regex
                     }
                }
            }
        }
    }
```

Use the `set` form of this command to configure a community list to define the match criteria for a community list rule, which is based on a regular expression for a community list.

Packets are matched based on whether the communities listed in the packet match the regular expression that is defined by using this command. Depending on the action that is defined for the rule by using policy route community-list [standard | expanded] { list-num | list-name } rule <rule-num> action *(page 45)*, matched packets are either permitted or denied.

Use the `delete` form of this command to delete the regular expression for a rule. If no regular expression is defined, all packets are considered to match the rule.

Use the `show` form of this command to display the regular expression for a rule.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route extcommunity-list [standard | expanded] { <list-num> | <list-name> } rule <rule-num> action

Specifies the action to take when packets match an extended community list rule.

Syntax:

```
set policy route extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num action
{ deny | permit }
```

Syntax:

```
delete policy route extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num
action
```

Syntax:

```
show policy route extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num
action
```

Packets that match this rule are forwarded.

list-num
: The number of a defined community list.

  A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name
: A string identifier for the community list.

  The string is a set of characters.

rule-num
: The rule number for a defined community list.

deny
: Silently drops the packets that match.

permit
: Forward packets that match the rule.

Configuration mode

```
policy {
    route {
        extcommunity-list {
                standard [list-num | list-name ]
                expanded [list-num | list-name ]
                    {
                         rule rule-num {
                            action {
```

```
                                deny
                                permit
                                }
                            }
                        }
                    }
            }
}
```

Use the `set` form of this command to define the action to specify the action to take when packets match an extended community list rule.

If the action for a rule is **`deny`**, packets that match the criteria of the rule are silently dropped. If the action for the rule is **`permit`**, destination-based routing is performed; that is, packets are sent by using the normal forwarding channels.

Use the `delete` form of this command to restore the default action to take for packets that match the criteria for a rule.

Use the `show` form of this command to display the action to take for a rule.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route extcommunity-list [standard | expanded] { <list-num> | <list-name> } rule <rule-num> description <desc>

Specifies a brief description of an extended community list rule.

Syntax:

```
set policy route extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num
description desc
```

Syntax:

```
delete policy extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num
description
```

Syntax:

```
show policy extcommunity-list [ standard | expanded ] { list-num | list-name } rule rule-num description
```

list-num

The number of a defined community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name

A string identifier for the community list.

The string is a set of characters.

rule-num

The rule number of a defined community list.

desc

A brief description for the community list rule.

Configuration mode

```
policy {
    extcommunity-list {
        standard [list-num | list-name ]
```

```
         expanded [list-num | list-name ]
            {
                 rule rule-num {
                      description desc
                            }
                }
        }
}
```

Use the `set` form of this command to create a description of an extended community list rule.

Use the `delete` form of this command to remove the description of an extended community list.

Use the `show` form of this command to display the description of an extended community list rule.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route extcommunity-list expanded { <list-num> | <list-name> } rule <rule-num> regex <regex>

Configures an extended community list to define the match criteria for a community list rule, which is based on a regular expression for a community list.

Syntax:

```
set policy route extcommunity-list expanded { list-num | list-name } rule rule-num regex regex
```

Syntax:

```
delete policy route extcommunity-list expanded { list-num | list-name } rule rule-num regex
```

Syntax:

```
show policy route extcommunity-list expanded { list-num | list-name } rule rule-num regex
```

If no regular expression is defined, all packets are considered to match the rule.

list-num

The number of a defined extended community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name

A string identifier for the extended community list.

The string is a set of characters.

rule-num

The number of a defined community list rule.

regex

A POSIX-style regular expression that represents a BGP community list.

Configuration mode

```
policy {
    route {
       extcommunity-list {
              expanded [list-num | list-name
                  {
                     rule rule-num {
                        regex regex
                           }
                   }
              }
         }
```

```
        }
```

Use the `set` form of this command to configure an expanded community list to define the match criteria for a community list rule, which is based on a regular expression for a community list.

Packets are matched based on whether the communities listed in the packet match the regular expression that is defined by using this command. Depending on the action that is defined for the rule by using policy route community-list [standard | expanded] { list-num | list-name } rule <rule-num> action *(page 45)*, matched packets are either permitted or denied.

Use the `delete` form of this command to delete the regular expression for a rule. If no regular expression is defined, all packets are considered to match the rule.

Use the `show` form of this command to display the regular expression for a rule.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route extcommunity-list standard { <list-num> | <list-name> } rule <rule-num> rt <route-target>

Configures an extended community list with a route target.

Syntax:

```
set policy route extcommunity-list standard { list-num | list-name } rule rule-num rt route-target
```

Syntax:

```
delete policy route extcommunity-list standard { list-num | list-name } rule rule-num rt route-target
```

Syntax:

```
show policy route extcommunity-list standard { list-num | list-name } rule rule-num rt route-target
```

list-num

The number of a defined extended community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

list-name

A string identifier for the extended community list.

The string is a set of characters.

rule-num

The rule number of defined extended community list.

route-target

A route target for an extended community list in either the AA:NN or IPaddress:NN format.

Configuration mode

```
policy {
    route {
        extcommunity-list {
                standard [list-num | list-name]
                 {
                    rule rule-num {
                        rt route-target
                           }
                 }
            }
        }
    }
```

Use the `set` form of this command to configure an extended community list with a route target.

Use the `delete` form of this command to delete an extended community list with a route target.

Use the `show` form of this command to display an extended community list with a route target.

Note: For more information about BGP community list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route extcommunity-list standard { <list-num> | <list-name> } rule <rule-num> soo <site-of-origin>

Configures an extended community list with a site of origin.

Syntax:

`set policy route extcommunity-list` **`standard`** { *list-num* | *list-name* } **`rule`** *rule-num* **`soo`** *site-of-origin-value*

Syntax:

`delete policy route extcommunity-list` **`standard`** { *list-num* | *list-name* } **`rule`** *rule-num* **`soo`** *site-of-origin-value*

Syntax:

`show policy route extcommunity-list` **`standard`** { *list-num* | *list-name* } **`rule`** *rule-num* **`soo`** *site-of-origin-value*

list-num

The number of a defined extended community list.

A standard community list number ranges from 1 through 99 and an expanded community list number ranges from 100 through 199.

rule-num

The rule number of a defined extended-community list.

site-of-origin-value

A site-of-origin for an extended community list in either the *AA:NN* or *IPaddress:NN* format.

Configuration mode

```
policy {
   route {
      extcommunity-list {
             standard [list-num | list-name]
              {
                rule rule-num {
                   soo site-of-origin-value
                 }
              }
           }
        }
    }
```

Use the `set` form of this command to configure an extended community list with site-of-origin.

Use the `delete` form of this command to delete an extended community list with site-of-origin.

Use the `show` form of this command to display an extended community list with a site-of-origin.

Note: For more information about BGP community-list, see the *"BGP Communities"* section in AT&T Vyatta Network Operating System BGP Configuration Guide.

policy route prefix-list <list-name>

Defines a prefix list.

Syntax:
`set policy route prefix-list` *list-name*

Syntax:
`delete policy route prefix-list` *list-name*

Syntax:
`show policy route prefix-list` *list-name*

list-name
Multi-node. A text identifier for the prefix list.

You can create multiple prefix lists by creating multiple **`policy route prefix-list`** configuration nodes.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        }
     }
}
```

Use the `set` form of this command to create a prefix list for use in policy-based routing.

Use the `delete` form of this command to remove a prefix list.

Use the `show` form of this command to display prefix list configuration.

policy route prefix-list <list-name> description <desc>

Allows you to specify a brief description for a prefix list.

Syntax:
`set policy route prefix-list` *list-name* **`description`** *desc*

Syntax:
`delete policy route prefix-list` *list-name* **`description`**

Syntax:
`show policy route prefix-list` *list-name* **`description`**

list-name
The name of a defined prefix list.

desc
A brief text description for the prefix list.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        description desc
         }
    }
}
```

Use the `set` form of this command to create a description for a prefix list.

Use the `delete` form of this command to remove a prefix list description.

Use the `show` form of this command to display the description for a prefix list.

policy route prefix-list <list-name> rule <rule-num>

Creates a rule for a prefix list.

Syntax:
`set policy route prefix-list` *list-name* **`rule`** *rule-num*

Syntax:
`delete policy route prefix-list` *list-name* **`rule`** *rule-num*

Syntax:
`show policy route prefix-list` *list-name* **`rule`** *rule-num*

list-name
The name of a defined prefix list.

rule-num
Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        rule rule-num {
         }
       }
    }
}
```

Use the `set` form of this command to create a prefix list rule.

Use the `delete` form of this command to remove a prefix list rule.

Use the `show` form of this command to display configuration settings for a prefix list rule.

policy route prefix-list <list-name> rule <rule-num> action

Specifies the action to be taken for packets matching a prefix list rule.

Syntax:
`set policy route prefix-list` *list-name* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:
`delete policy route prefix-list` *list-name* **`rule`** *rule-num* **`action`**

Syntax:
`show policy route prefix-list` *list-name* **`rule`** *rule-num* **`action`**

Packets matching this rule are forwarded.

list-name
The name of a defined prefix list.

rule-num
The number of a defined prefix list rule.

deny
Packets matching this rule are silently dropped.

permit
Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        rule rule-num {
            action {
                deny
                permit
             }
          }
       }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **deny**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **permit**, destination-based routing is performed; that is, packets are sent using the normal forwarding channels.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route prefix-list <list-name> rule <rule-num> description <desc>

Allows you to specify a brief description for a prefix list rule.

Syntax:
`set policy route prefix-list` *list-name* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route prefix-list` *list-name* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route prefix-list` *list-name* **`rule`** *rule-num* **`description`**

list-name
The name of a defined prefix list.

rule-num
The number of a defined prefix list rule.

desc
A brief text description for the prefix list rule.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        rule rule-num {
            description desc
          }
       }
    }
```

```
}
```

Use the `set` form of this command to create a description for a prefix list rule.

Use the `delete` form of this command to remove a prefix list rule description.

Use the `show` form of this command to display the description for a prefix list rule.

policy route prefix-list <list-name> rule <rule-num> ge <value>

Defines match criteria for a prefix list rule based on a “greater-than-or-equal-to” numeric comparison.

Syntax:

`set policy route prefix-list` *list-name* **`rule`** *rule-num* **`ge`** *value*

Syntax:

`delete policy route prefix-list` *list-name* **`rule`** *rule-num* **`ge`**

Syntax:

`show policy route prefix-list` *list-name* **`rule`** *rule-num* **`ge`**

If no prefix is specified, all network prefixes are considered to match the rule.

list-name
The name of a defined prefix list.

rule-num
The number of a defined prefix list rule.

value
A number representing a network prefix. Network prefixes greater than or equal to this number will match this rule. The range of values is 0 to 32.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        rule rule-num {
            ge value
          }
        }
   }
}
```

Use the `set` form of this command to specify a network prefix for determining routing. The network prefixes of incoming packets are compared with this value; if the prefix is greater than or equal to the specified prefix, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified “ge” prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as “ge” prefix.

policy route prefix-list <list-name> rule <rule-num> le <value>

Defines a match criterion based on a “less-than-or-equal-to” numeric comparison for a prefix list rule.

Syntax:

```
set policy route prefix-list list-name rule rule-num le value
```

Syntax:

```
delete policy route prefix-list list-name rule rule-num le
```

Syntax:

```
show policy route prefix-list list-name rule rule-num le
```

If no prefix is specified, all network prefixes are considered to match the rule.

list-name
The name of a defined prefix list.

rule-num
The number of a defined prefix list rule.

value
A number representing a network prefix. Network prefixes less than or equal to this number will match this rule. The range of values is 0 to 32.

Configuration mode

```
policy {
    route {
      prefix-list list-name {
        rule rule-num {
            le value
           }
        }
    }
}
```

Use the `set` form of this command to specify a network prefix for determining routing policy. The network prefixes of incoming packets are compared with this value; if the prefix is less than or equal to the specified prefix, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified "le" prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as "le" prefix.

policy route prefix-list <list-name> rule <rule-num> prefix <ipv4net>

Defines match criteria for a prefix list rule based on an IPv4 network.

Syntax:

```
set policy route prefix-list list-name rule rule-number prefix ipv4net
```

Syntax:

```
delete policy route prefix-list list-name rule rule-num prefix
```

Syntax:

```
show policy route prefix-list list-name rule rule-num prefix
```

If no network is specified, all networks are considered to match the rule.

list-name
The name of a defined prefix list.

rule-num
The number of a defined prefix list rule.

ipv4net

An IPv4 network. Networks exactly matching this network will match this rule. The format is *ip-address/prefix*.

Configuration mode

```
policy {
    route {
       prefix-list list-name {
          rule rule-number {
            prefix ipv4net
           }
        }
    }
}
```

Use the `set` form of this command to specify a network for determining routing policy. The network specified in incoming packets are compared with this value; if it exactly matches the network specified in this command, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified "ge" prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as "ge" prefix.

policy route prefix-list6 <list-name>

Defines an IPv6 prefix list.

Syntax:
`set policy route prefix-list6` *list-name*

Syntax:
`delete policy route prefix-list6` *list-name*

Syntax:
`show policy route prefix-list6` *list-name*

list-name

Multi-node. A text identifier for the IPv6 prefix list.

You can create multiple IPv6 prefix lists by creating multiple **`policy route prefix-list6`** configuration nodes.

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        }
    }
}
```

Use the `set` form of this command to create a prefix list for use in policy-based routing.

Use the `delete` form of this command to remove a prefix list.

Use the `show` form of this command to display prefix list configuration.

policy route prefix-list6 <list-name> description <desc>

Allows you to specify a brief description for an IPv6 prefix list.

Syntax:
`set policy route prefix-list6` *list-name* **`description`** *desc*

Syntax:
`delete policy route prefix-list6` *list-name* **`description`**

Syntax:
`show policy route prefix-list6` *list-name* **`description`**

list-name
The name of a defined IPv6 prefix list.

desc
A brief text description for the prefix list.

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        description desc
        }
    }
}
```

Use the `set` form of this command to create a description for a prefix list.

Use the `delete` form of this command to remove a prefix list description.

Use the `show` form of this command to display the description for a prefix list.

policy route prefix-list6 <list-name> rule <rule-num>

Creates a rule for an IPv6 prefix list.

Syntax:
`set policy route prefix-list6` *list-name* **`rule`** *rule-num*

Syntax:
`delete policy route prefix-list6` *list-name* **`rule`** *rule-num*

Syntax:
`show policy route prefix-list6` *list-name* **`rule`** *rule-num*

list-name
The name of a defined IPv6 prefix list.

rule-num
Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        rule rule-num {
```

```
        }
      }
    }
}
```

Use the `set` form of this command to create a prefix list rule.

Use the `delete` form of this command to remove a prefix list rule.

Use the `show` form of this command to display configuration settings for a prefix list rule.

policy route prefix-list6 <list-name> rule <rule-num> action

Specifies the action to be taken for packets matching an IPv6 prefix list rule.

Syntax:

`set policy route prefix-list6` *list-name* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:

`delete policy route prefix-list6` *list-name* **`rule`** *rule-num* **`action`**

Syntax:

`show policy route prefix-list6` *list-name* **`rule`** *rule-num* **`action`**

Packets matching this rule are forwarded.

list-name
The name of a defined IPv6 prefix list.

rule-num
The number of a defined IPv6 prefix list rule.

`deny`
Packets matching this rule are silently dropped.

`permit`
Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        rule rule-num {
            action {
                deny
                permit
             }
          }
      }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **`deny`**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **`permit`**, destination-based routing is performed; that is, packets are sent using the normal forwarding channels.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route prefix-list6 <list-name> rule <rule-num> description <desc>

Allows you to specify a brief description for an IPv6 prefix list rule.

Syntax:
`set policy route prefix-list6` *list-name* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route prefix-list6` *list-name* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route prefix-list6` *list-name* **`rule`** *rule-num* **`description`**

list-name
The name of a defined IPv6 prefix list.
rule-num
The number of a defined IPv6 prefix list rule.
desc
A brief text description for the prefix list rule.

Configuration mode

```
policy {
    route {
       prefix-list6 list-name {
        rule rule-num {
            description desc
          }
       }
    }
}
```

Use the `set` form of this command to create a description for a prefix list rule.

Use the `delete` form of this command to remove a prefix list rule description.

Use the `show` form of this command to display the description for a prefix list rule.

policy route prefix-list6 <list-name> rule <rule-num> ge <value>

Defines match criteria for an IPv6 prefix list rule based on a “greater-than-or-equal-to” numeric comparison.

Syntax:
`set policy route prefix-list6` *list-name* **`rule`** *rule-num* **`ge`** *value*

Syntax:
`delete policy route prefix-list6` *list-name* **`rule`** *rule-num* **`ge`**

Syntax:
`show policy route prefix-list6` *list-name* **`rule`** *rule-num* **`ge`**

If no prefix is specified, all network prefixes are considered to match the rule.

list-name
The name of a defined IPv6 prefix list.
rule-num
The number of a defined IPv6 prefix list rule.

value

A number representing a network prefix. Network prefixes greater than or equal to this number will match this rule. The range of values is 0 to 128.

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        rule rule-num {
            ge value
          }
        }
    }
}
```

Use the `set` form of this command to specify a network prefix for determining routing. The network prefixes of incoming packets are compared with this value; if the prefix is greater than or equal to the specified prefix, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified "ge" prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as "ge" prefix.

policy route prefix-list6 <list-name> rule <rule-num> le <value>

Defines a match criterion based on a "less-than-or-equal-to" numeric comparison for an IPv6 prefix list rule.

Syntax:

`set policy route prefix-list6` *list-name* **`rule`** *rule-num* **`le`** *value*

Syntax:

`delete policy route prefix-list6` *list-name* **`rule`** *rule-num* **`le`**

Syntax:

`show policy route prefix-list6` *list-name* **`rule`** *rule-num* **`le`**

If no prefix is specified, all network prefixes are considered to match the rule.

list-name

The name of a defined IPv6 prefix list.

rule-num

The number of a defined IPv6 prefix list rule.

value

A number representing a network prefix. Network prefixes less than or equal to this number will match this rule. The range of values is 0 to 128.

Configuration mode

```
policy {
    route {
     prefix-list6 list-name {
        rule rule-num {
            le value
          }
        }
    }
```

```
}
```

Use the `set` form of this command to specify a network prefix for determining routing policy. The network prefixes of incoming packets are compared with this value; if the prefix is less than or equal to the specified prefix, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified "le" prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as "le" prefix.

policy route prefix-list6 <list-name> rule <rule-num> prefix <ipv6net>

Defines match criteria for a prefix list rule based on an IPv6 network.

Syntax:

`set policy route prefix-list6` *list-name* **`rule`** *rule-number* **`prefix`** *ipv6net*

Syntax:

`delete policy route prefix-list6` *list-name* **`rule`** *rule-num* **`prefix`**

Syntax:

`show policy route prefix-list6` *list-name* **`rule`** *rule-num* **`prefix`**

If no network is specified, all networks are considered to match the rule.

list-name
: The name of a defined prefix list.

rule-num
: The number of a defined prefix list rule.

ipv6net
: An IPv6 network. Networks exactly matching this network will match this rule. The format is *ipv6-address/prefix* (that is *x:x:x:x:x:x:x:x*/0-128).

Configuration mode

```
policy {
    route {
      prefix-list6 list-name {
        rule rule-number {
            prefix ipv6net
         }
       }
    }
}
```

Use the `set` form of this command to specify a network for determining routing policy. The network specified in incoming packets are compared with this value; if it exactly matches the network specified in this command, the rule is matched and the action specified for the rule is taken.

Exactly one comparison (**`ge`**, **`le`**, or **`prefix`**) may be specified for a prefix list rule.

Use the `delete` form of this command to remove the specified "ge" prefix. If no prefix is specified, all network prefixes are considered to match the rule.

Use the `show` form of this command to display the value specified as "ge" prefix.

policy route route-map <map-name>

Defines a route map for policy-based routing.

Syntax:
`set policy route route-map` *map-name*

Syntax:
`delete policy route route-map` *map-name*

Syntax:
`show policy route route-map` *map-name*

map-name
Multi-node. A text identifier for the route map.

You can create multiple route maps by creating multiple **`policy route route-map`** configuration nodes.

Configuration mode

```
policy {
    route-map map-name {}
}
```

Use the `set` form of this command to create a route map for policy-based routing.

Use the `delete` form of this command to remove a route map.

Use the `show` form of this command to display route map configuration.

policy route route-map <map-name> description <desc>

Allows you to specify a brief description for a route map.

Syntax:
`set policy route route-map` *map-name* **`description`** *desc*

Syntax:
`delete policy route route-map` *map-name* **`description`**

Syntax:
`show policy route route-map` *map-name* **`description`**

map-name
The name of a defined route map.

desc
A brief text description for the route map.

Configuration mode

```
policy {
    route-map map-name {
        description desc
    }
}
```

Use the `set` form of this command to create a description for a route map.

Use the `delete` form of this command to remove a route map policy description.

Use the `show` form of this command to display the description for a route map.

policy route route-map <map-name> rule <rule-num>

Creates a rule for a route map.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num*

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num*

map-name
The name of a defined route map.

rule-num
Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.

You can define multiple rules by creating multiple **`rule`** configuration nodes.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {}
    }
}
```

Use the `set` form of this command to create a route map rule.

Use the `delete` form of this command to remove a route map rule.

Use the `show` form of this command to display configuration settings for a route map rule.

Note: Apply the route-map to neighbor for the policies to take affect.

policy route route-map <map-name> rule <rule-num> action

Specifies the action to be taken for packets matching a route map rule.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`action`** { **`deny`** | **`permit`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`action`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`action`**

Routes are denied.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`deny`
Packets matching this rule are silently dropped.

`permit`
Packets matching this rule are forwarded.

Configuration mode

```
policy {
    route-map map-name {
```

```
        rule rule-num {
            action {
                deny
                permit
            }
        }
    }
}
```

Use the `set` form of this command to define the action taken when received packets satisfy the match criteria for this rule.

If the action for a rule is **`deny`**, packets meeting the match criteria of the rule are silently dropped. If the action for the rule is **`permit`**, destination-based routing is performed; that is, packets are sent using the normal forwarding channels.

The default action of a route map is to deny; that is, if no entries satisfy the match criteria, the route is denied. To change this behavior, specify an empty **`permit`** rule as the last entry in the route map.

Use the `delete` form of this command to restore the default action for packets satisfying the match criteria.

Use the `show` form of this command to display action settings for this rule.

policy route route-map <map-name> rule <rule-num> continue <target-num>

Calls to another rule within the current route map.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`continue`** *target-num*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`continue`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`continue`**

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

target
The identifier of the route map rule being called.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            continue target-num
        }
    }
}
```

Use the `set` form of this command to call to another rule within the current route map. The new route map rule is called after all **`set`** actions specified in the route map rule have been performed.

Use the `delete` form of this command to remove this statement from the route map.

Use the `show` form of this command to display route map rule configuration settings.

policy route route-map <map-name> rule <rule-num> description <desc>

Allows you to specify a brief description for a route map rule.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`description`** *desc*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`description`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`description`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
desc
A brief text description for the route map rule.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            description desc
        }
    }
}
```

Use the `set` form of this command to create a description for a route map rule.

Use the `delete` form of this command to remove a route map rule description.

Use the `show` form of this command to display the description for a route map rule.

policy route route-map <map-name> rule <rule-num> match as-path <list-name>

Defines a match condition for a route map based on an AS path list.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match as-path`** *list-name*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match as-path`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`match as-path`**

If no AS path match condition is specified, packets are not filtered by AS path.

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
list-name

Matches the AS paths in the route with those permitted by the specified AS path list. The AS path list must already be defined.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                as-path list-name
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on an AS path list.

Packets are matched based on whether the AS path listed in the route match the AS path defined by this command. Depending on the action defined for the rule using policy route route-map map-name rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the AS path match condition.

Use the `show` form of this command to display AS path match condition configuration.

policy route route-map <map-name> rule <rule-num> match community

Defines a match condition for a route map based on BGP communities.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match community`** { **`community-list`** *list-num* | **`exact-match`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match community`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`match community`**

If no community list match condition is specified, packets are not filtered by BGP community.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`community-list` ***list-num***
Matches the BGP communities in the route with those permitted by the specified community list. The community list policy must already be defined. Either **`community-list`** or **`exact-match`** must be specified.

`exact-match`
BGP communities are to be matched exactly. Either **`community-list`** or **`exact-match`** must be specified.

Configuration mode

```
policy {
    route-map map-name {
```

```
        rule rule-num {
            match {
                community {
                    community-list list-num
                    exact-match
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on BGP communities.

Packets are matched based on whether the BGP communities listed in the route match the communities defined by this command. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the BGP community match condition.

Use the `show` form of this command to display BGP community match condition configuration.

policy route route-map <map-name> rule <rule-num> match extcommunity

Defines a match condition for a route map based on BGP extended communities.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match extcommunity`** { **`community-list`** *list-num* | **`exact-match`** }

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match extcommunity`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match extcommunity`**

If no community list match condition is specified, packets are not filtered by BGP extended community.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`extcommunity-list` ***list-num***
Matches the BGP extended communities in the route with those permitted by the specified community list. The community list policy must already be defined. Either **`extcommunity-list`** or **`exact-match`** must be specified.

`exact-match`
BGP communities are to be matched exactly. Either **`extcommunity-list`** or **`exact-match`** must be specified.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
```

```
                extcommunity {
                    extcommunity-list list-num
                    exact-match
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on BGP extended communities.

Packets are matched based on whether the BGP communities listed in the route match the communities defined by this command. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **`set`** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the BGP extended community match condition.

Use the `show` form of this command to display BGP extended community match condition configuration.

policy route route-map <map-name> rule <rule-num> match interface <interface-name>

Defines a match condition for a route map based on the first-hop interface.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match interface`** *interface-name*

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match interface`** *interface-name*

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match interface`** *interface-name*

If no interface match condition is specified, packets are not filtered by interface.

map-name
: The name of a defined route map.

rule-number
: The number of a defined route map rule.

interface-name
: Matches first hop interface specified in the route against the interface name.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                interface interface-name
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on first-hop interface.

Packets are matched based on whether the first-hop interface of the route matches the interface specified by this command. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the interface match condition.

Use the `show` form of this command to display interface match condition configuration.

policy route route-map <map-name> rule <rule-num> match ip address

Defines a match condition for a route map based on IP address.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ip address`** { **`access-list`** *list-num* | **`prefix-list`** *list-name* }

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match ip address`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match ip address`**

If no IP address match condition is specified, packets are not filtered by IP address.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`access-list` ***list-num***
Matches the source or destination IP address of the route against those permitted by the specified access list. The access list must already be defined. Either **`access-list`** or **`prefix-list`** must be specified.

`prefix-list` ***list-name***
Matches the source or destination network of the route against those permitted by the specified prefix list. The prefix list must already be defined. Either **`access-list`** or **`prefix-list`** must be specified.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                ip address {
                        access-list list-num
                        prefix-list list-name
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on IP address.

Packets are matched based on whether the source or destination IP address of the route matches an address contained in the specified access list or prefix list. Depending on the action defined for the rule using policy

route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the IP address match condition.

Use the `show` form of this command to display IP address match condition configuration.

policy route route-map <map-name> rule <rule-num> match ip nexthop

Defines a match condition for a route map based on the next-hop address.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ip nexthop`** { **`access-list`** *list-num* | **`prefix-list`** *list-name* }

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match ip nexthop`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match ip nexthop`**

If no next-hop match condition is specified, packets are not filtered by next hop.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`access-list` ***list-num***
Matches the next-hop IP address in the route against those permitted by the specified access list. The access list must already be defined. Either **`access-list`** or **`prefix-list`** must be specified.

`prefix-list` ***list-name***
Matches next-hop IP address in the route against those permitted by the specified prefix list. The prefix list must already be defined. Either **`access-list`** or **`prefix-list`** must be specified.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                ip {
                    nexthop {
                        access-list list-num
                        prefix-list list-name
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on next-hop IP address.

Packets are matched based on whether the next-hop IP address of the route matches an address contained in the specified access list or prefix list. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based

on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the next-hop IP address match condition.

Use the `show` form of this command to display next-hop IP address match condition configuration.

policy route route-map <map-name> rule <rule-num> match ip peer access-list <list-num>

Defines a match condition for a route map based on a list.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ip peer access-list`** *list-num*

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match ip peer access-list`** *list-num*

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match ip peer`**

If no list is specified, packets are not filtered by IP address.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`access-list` ***list-num***
Matches the source or destination IP address of the route against those permitted by the specified access list. The access list must already be defined.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                ip {
                    peer {
                            access-list list-num
                        }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map based on a list.

Packets are matched based on whether the source or destination IP address of the route matches an address contained in the specified access list .

Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the IP list match condition.

Use the `show` form of this command to display IP list match condition configuration.

policy route route-map <map-name> rule <rule-num> match ipv6 address

Defines a match condition for a route map based on IPv6 address.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ipv6 address`** { **`access-list6`** *list-num* | **`prefix-list6`** *list-name* }

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match ipv6 address`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`match ipv6 address`**

If no IPv6 address match condition is specified, packets are not filtered by IPv6 address.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`access-list6` ***list-num***
Matches the source or destination IP address of the route against those permitted by the specified access list. The access list must already be defined. Either **`access-list6`** or **`prefix-list6`** must be specified.

`prefix-list6` ***list-name***
Matches the source or destination network of the route against those permitted by the specified prefix list. The prefix list must already be defined. Either **`access-list6`** or **`prefix-list6`** must be specified.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                ipv6 address {
                        access-list6 list-num
                        prefix-list6 list-name

                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based on IPv6 address.

Packets are matched based on whether the source or destination IPv6 address of the route matches an address contained in the specified access list or prefix list. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **`set`** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the IPv6 address match condition.

Use the `show` form of this command to display IPv6 address match condition configuration.

policy route route-map <map-name> rule <rule-num> match ipv6 nexthop

Defines a match condition for a route map based on the next-hop IPv6 address.

Syntax:

```
set policy route route-map map-name rule rule-num match ipv6 nexthop { access-list6 list-num | prefix-list6 list-name }
```

Syntax:

```
delete policy route route-map map-name rule rule-num match ipv6 nexthop
```

Syntax:

```
show policy route route-map map-name rule rule-num match ipv6 nexthop
```

If no next-hop match condition is specified, packets are not filtered by next hop.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

access-list6 ***list-num***
Matches the next-hop IPv6 address in the route against those permitted by the specified access list. The access list must already be defined. Either **access-list6** or **prefix-list6** must be specified.

prefix-list6 ***list-name***
Matches next-hop IPv6 address in the route against those permitted by the specified prefix list. The prefix list must already be defined. Either **access-list6** or **prefix-list6** must be specified.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                ipv6 {
                    nexthop {
                        access-list6 list-num
                        prefix-list6 list-name
                    }
                }
            }
        }
    }
}
```

Use the **set** form of this command to define a match condition for a route map policy based on next-hop IPv6 address.

Packets are matched based on whether the next-hop IPv6 address of the route matches an address contained in the specified access list or prefix list. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **set** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the **delete** form of this command to remove the next-hop IPv6 address match condition.

Use the **show** form of this command to display next-hop IPv6 address match condition configuration.

policy route route-map <map-name> rule <rule-num> match metric <metric>

Defines a match condition for a route map based on the route's metric.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match metric`** *metric*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match metric`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`match metric`**

If no metric match condition is specified, packets are not filtered by metric.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

metric
A number representing a route metric. This value is matched against the metric in the route.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                metric metric
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based route metric.

Packets are matched based on whether the route metric matches that specified by this command. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **`set`** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the route source match condition.

Use the `show` form of this command to display route source match condition configuration.

policy route route-map <map-name> rule <rule-num> match origin

Defines a match condition for a route map based on the route's origin.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match origin`** { **`egp`** | **`igp`** | **`incomplete`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match origin`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`match origin`**

If no origin match condition is specified, packets are not filtered by BGP origin code.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

`egp`
Matches routes whose origin is an Exterior Gateway Protocol.

`igp`
Matches routes whose origin is an Interior Gateway Protocol.

`incomplete`
Matches routes whose BGP origin code is incomplete.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                origin {
                    origin-code [egp|igp|incomplete]
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a match condition for a route map policy based BGP origin.

Packets are matched based on whether the BGP origin code in the route matches that specified by this command. Depending on the action defined for the rule using policy route route-map <map-name> rule <rule-num> action *(page 64)*, matched packets are either permitted or denied. Based on the forwarding information specified by the **`set`** statements in the route map rule, permitted packets are forwarded to their various destinations.

If more than one match condition is defined in a route map rule, the packet must match all conditions to count as a match. If no match condition is defined for the route map rule, all packets are considered to match the rule.

Use the `delete` form of this command to remove the origin match condition.

Use the `show` form of this command to display origin match condition configuration.

policy route route-map <map-name> rule <rule-num> match tag <tag>

Defines a match condition for a route map based on OSPF tag.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match tag`** *tag*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`match tag`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`match tag`**

If no tag match condition is specified, packets are not filtered by tag.

map-name

The name of a defined route map.

rule-num

The number of a defined route map rule.

tag

A 32-bit value representing an OSPF tag. This value is matched against the contents of the OSPF external Link-State Advertisement (LSA) 32-bit tag field in the route.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            match {
                tag tag
            }
        }
    }
}
```

Use the `set` form of this command to define an exit policy for a route map entry, by specifying the route map rule to be executed when a match occurs. When all the match conditions specified by the route map rule succeed, the route map rule specified by this command is invoked and executed.

Normally, when a route map is matched, the route map is exited and the route is permitted. This command allows you to specify an alternative exit policy, by directing execution to a specified route map rule or to the next rule in the sequence.

Use the `delete` form of this command to remove the exit policy.

Use the `show` form of this command to display route map exit policy configuration.

policy route route-map <map-name> rule <rule-num> set aggregator

Modifies the BGP aggregator attribute of a route.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`set aggregator`** { **`as`** *asn* | **`ip`** *ipv4* }

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set aggregator`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name

The name of a defined route map.

rule-num

The number of a defined route map rule.

`as` ***asn***

Modifies the autonomous system number of the BGP aggregator in the route to the specified value. The range is 1 to 65535.

`ip` ***ipv4***

Modifies the IP address of the BGP aggregator in the route to the specified IPv4 address.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
```

```
                aggregator {
                    as asn
                    ip ipv4
                }
            }
        }
    }
}
```

Use the `set` form of this command to modify the aggregator attribute of a route. When all the match conditions in the route map rule succeed, the aggregator attribute is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set as-path-prepend <prepend>

Sets or prepends to the AS path of the route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set as-path-prepend`** *prepend*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set as-path-prepend`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
prepend
A string representing an AS path.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                as-path-prepend prepend
            }
        }
    }
}
```

Use the `set` form of this command to prepend a string to the AS path list in a route. When all the match conditions in the route map rule succeed, the specified string is prepended to the AS path in the route.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set atomic-aggregate

Sets the BGP atomic-aggregate attribute in a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set atomic-aggregate`**

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set atomic-aggregate`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                atomic-aggregate
            }
        }
    }
}
```

Use the `set` form of this command to set the BGP atomic aggregate attribute in a route. When all the match conditions in the route map rule succeed, the BGP atomic aggregate attribute is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set community

Modifies the BGP community list in a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set community`** [*AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`**]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set community`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
aa:nn
Specifies the community in 4-octet, AS-value format.
`local-AS`
Advertises communities in local AS only (NO_EXPORT_SUBCONFED).
`no-advertise`
Does not advertise this route to any peer (NO_ADVERTISE).

no-export
Does not advertise outside of this AS of confederation boundary (NO_EXPORT).

internet
Specifies the 0 symbolic Internet community.

none
Specifies no communities.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                community AA:NN
                    local-AS
                    no-advertise
                    no-export
                    internet
                    none
            }
        }
    }
}
```

Use the `set` form of this command to modify the BGP community list in a route. When all the match conditions in the route map rule succeed, the community list is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

Note: The community list must already be defined.

policy route route-map <map-name> rule <rule-num> set add-community <community>

Adds a BGP community to an existing community.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`action`** [**`permit`** | **`deny`**]

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ip address prefix-list`** *prefix-num*

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set add-community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set add-community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** }

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set add-community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** }

Syntax:

map-name
The name of a defined route map.

list-num
The number of a defined community list.
rule-num
The number of a defined community list rule.
aa:nn
Specifies the community in 4-octet, AS-value format.
local-AS
Advertises communities in local AS only. (NO_EXPORT_SUBCONFED).
no-advertise
Does not advertise this route to any peer (NO_ADVERTISE).
no-export
Does not advertise outside of this AS of confederation boundary. (NO_EXPORT).
internet
Specifies the 0 symbolic Internet community.
none
Specifies no communities.

Configuration mode

```
policy {
    route {
        route-map map-name {
            rule rule-num {
                action {
                        deny
                        permit
                match {
                    ip {
                        address {
                            prefix-list prefix-num {
                                set {
                                     add-community AA:NN
                                     local-AS
                                     no-advertise
                                     no-export
                                     internet
                                     none
                                     }
                                 }
                             }
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to add a BGP community to an existing community.

Use the `delete` form of this command to delete the newly added BGP community from an existing community.

Use the `show` form of this command to display the configuration for route maps.

Note: You cannot configure this command and `set policy route route-map` *map-name* **`rule`** *rule-num* **`set community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** } at the same time.

policy route route-map <map-name> rule <rule-num> set add-extcommunity rt <community>

Adds a BGP extended community to an existing extended community.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set add-extcommunity rt`** { *AA:NN* | *IPAddr-NN* }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set add-extcommunity rt`** { *AA:NN* | *IPAddr-NN* }

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set add-extcommunity rt`** { *AA:NN* | *IPAddr-NN* }

map-name
The name of a defined route map.
rule-num
The number of a defined extended community list rule.
AA:NN
An extended community in 4-octet, AS-value format.
IPAddr-NN
An extended community in IP address-NN format.

Configuration mode

```
policy {
    route {
        route-map map-name {
            rule rule-num {
               set {
                     add-extcommunity {
                       rt {
                            AA:NN
                            IPAddr-NN

                       }
                     }
               }
            }
         }
      }
}
```

Use the `set` form of this command to add a BGP extended community to an existing extended community.

Use the `delete` form of this command to delete the newly added BGP extended community from an existing extended community.

Use the `show` form of this command to display the configuration for route maps.

policy route route-map <map-name> rule <rule-num> set community <community>

Modifies a BGP community only if it matches a prefix-list.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`action`** [**`permit`** | **`deny`**]

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`match ip address prefix-list`** *prefix-num*

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set community`** { *AA:NN* | **`local-AS`** | **`no-advertise`** | **`no-export`** | **`internet`** | **`none`** }

map-name
The name of a defined route map.
list-num
The number of a defined community list.
rule-num
The number of a defined community list rule.
aa:nn
Specifies the community in 4-octet, AS-value format.
local-AS
Advertises communities in local AS only (NO_EXPORT_SUBCONFED).
no-advertise
Does not advertise this route to any peer (NO_ADVERTISE).
no-export
Does not advertise outside of this AS of confederation boundary (NO_EXPORT).
internet
Specifies the 0 symbolic Internet community.
none
Specifies no communities.

Configuration mode

```
policy {
    route {
        route-map map-name {
            rule rule-num {
                action {
                        deny
                        permit
                match {
                    ip {
                        address {
                            prefix-list prefix-num {
                               set {
                                   community AA:NN
                                   local-AS
                                   no-advertise
                                   no-export
                                   internet
                                   none
                                   }
                               }
                           }
                        }
                }
            }
        }
    }
}
```

Use the `set` form of this command to to modify the BGP community attribute in a route.

Note: The community list must already be defined.

policy route route-map <map-name> rule <rule-num> set ext-community <community>

Modifies a BGP extended community only if it matches a prefix-list.

Syntax:

```
set policy route route-map map-name rule rule-num action [ permit | deny ]
```

Syntax:

```
set policy route route-map map-name rule rule-num match ip address prefix-list prefix-num
```

Syntax:

```
set policy route route-map map-name rule rule-num set extcommunity { AA:NN | local-AS | no-advertise | no-export | internet | none }
```

map-name
The name of a defined route map.

list-num
The number of a defined community list.

rule-num
The number of a defined community list rule.

aa:nn
Specifies the community in 4-octet, AS-value format.

local-AS
Advertises communities in local AS only (NO_EXPORT_SUBCONFED).

no-advertise
Does not advertise this route to any peer (NO_ADVERTISE).

no-export
Does not advertise outside of this AS of confederation boundary (NO_EXPORT).

internet
Specifies the 0 symbolic Internet community.

none
Specifies no communities.

Configuration mode

```
policy {
    route {
        route-map map-name {
            rule rule-num {
                action {
                        deny
                        permit
                match {
                    ip {
                            address {
                              prefix-list prefix-num {
                                 set { extcommunity AA:NN
                                           local-AS
                                           no-advertise
                                           no-export
                                           internet
                                           none
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to modify the BGP extended-community attribute in a route.

policy route route-map <map-name> rule <rule-num> set community <action>

Modifies the BGP communities attribute in a route.

Syntax:

```
set policy route route-map map-name rule rule-num set community { AA:NN | local-AS | no-advertise | no-export | internet | none }
```

Syntax:

```
delete policy route route-map map-name rule rule-num set community [ AA:NN | local-AS | no-advertise | no-export | internet | none ]
```

Syntax:

```
show policy route route-map map-name rule rule-num set community
```

When the **additive** keyword is not used, the specified community replaces the existing communities in the route.

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

aa:nn
Specifies the community in 4-octet, AS-value format.

local-AS
Advertises communities in local AS only (NO_EXPORT_SUBCONFED).

no-advertise
Does not advertise this route to any peer (NO_ADVERTISE).

no-export
Does not advertise outside of this AS of confederation boundary (NO_EXPORT).

internet
Specifies the 0 symbolic Internet community.

none
Specifies no communities.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                community
                    AA:NN
                    local-AS
                    no-advertise
                    no-export
                    internet
                    none
            }
        }
    }
}
```

Use the `set` form of this command to modify the BGP communities attribute in a route. When all the match conditions in the route map rule succeed, the communities attribute is modified as specified by the rule.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display set statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set delete-community <list-id-or-name>

Deletes a BGP community list from a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set delete-community`** { *list-id* | *name* }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set delete-community`** [*list-id* | *name*]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set delete-community`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
list-id
A community-list identifier, a number that ranges from 1 through 199.
name
The name of a community list.

Configuration mode

```
policy {
       route {
               route-map map-name {
                       rule rule-num {
                               set {
                                       delete-community list-id
                                       delete-community name
                               }
                       }
               }
       }
}
```

This command deletes a BGP community list from a route. The community list must already be defined.

Use the `set` form of this command to delete a BGP community list from a route.

Use the `delete` form of this command to undelete a BGP community list from a route.

Use the `show` form of this command to display the deleted community lists.

policy route route-map <map-name> rule <rule-num> set delete-extcommunity <list-id-or-name>

Deletes a BGP extended community list from a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set delete-extcommunity`** { *list-id* | *list-name* }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set delete-extcommunity`** [*list-id* | *list-name*]

Syntax:

```
show policy route route-map map-name rule rule-num set delete-extcommunity
```

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

list-id
An extended community-list identifier, a number that ranges from 1 through 199.

list-name
A configured extended community-list name.

Configuration mode

```
policy {
       route {
               route-map map-name {
                       rule rule-num {
                               set {
                                       delete-extcommunity list-id
                                       delete-extcommunity pattern
                               }
                       }
               }
       }
}
```

This command deletes a BGP extended community list from a route. The extended community list must already be defined.

Use the `set` form of this command to delete a BGP extended community list from a route.

Use the `delete` form of this command to undelete a BGP extended community list from a route.

Use the `show` form of this command to display the deleted extended community lists.

policy route route-map <map-name> rule <rule-num> set ip-next-hop <ipv4>

Modifies the next hop destination of a route.

Syntax:

```
set policy route route-map map-name rule rule-num set ip-next-hop ipv4
```

Syntax:

```
delete policy route route-map map-name rule rule-num set ip-next-hop [ ipv4 ]
```

Syntax:

```
show policy route route-map map-name rule rule-num set ip-next-hop
```

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

ipv4
The IPv4 address of the next hop.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
```

```
            set {
                ip-next-hop ipv4
            }
        }
    }
}
```

Use the `set` form of this command to modify the next hop destination for packets that traverse a route map. When all the match conditions in the route map rule succeed, the next hop of the route is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set ipv6-next-hop <scope> <ipv6>

Modifies the IPv6 next hop destination of a route.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`set ipv6-next-hop`** { **`global`** | **`local`** } *ipv6*

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set ipv6-next-hop`** [**`global`** | **`local`**]

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
: The name of a defined route map.

rule-num
: The number of a defined route map rule.

`global`
: The next hop address is an IPv6 global address.

`local`
: The next hop address is an IPv6 local address.

ipv6
: The IPv6 address of the next hop.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                ipv6-next-hop {
                    global ipv6
                    local ipv6
                }
            }
        }
    }
}
```

When all the match conditions in the route map rule succeed, the next hop of the route is modified as specified.

Use the `set` form of this command to modify the IPv6 next hop destination address for packets that traverse a route map.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set local-preference <local-pref>

Modifies the BGP local-pref attribute in a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set local-preference`** *local-pref*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set local-preference`** [*local-pref*]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set local-preference`**

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

local-pref
The new value for the BGP local preference path attribute. The numbers range from 0 through 4294967295.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                local-preference local-pref
            }
        }
    }
}
```

Use the `set` form of this command to modify the BGP local-pref attribute for packets that traverse a route map. When all the match conditions in the route map rule succeed, the local-pref attribute of the route is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set metric <metric>

Modifies the metric of a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set metric`** *metric*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set metric`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

metric
A number representing the new metric to be used in the route.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                metric metric
            }
        }
    }
}
```

Use the `set` form of this command to modify the route metric for packets that traverse a route map. When all the match conditions in the route map rule succeed, the route metric is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set metric-type <type>

Specifies the OSPF external metric-type for a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set metric-type`** [*type-1* | *type-2*]

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set metric-type`** [*type-1* | *type-2*]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set metric-type`**

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

type-1
OSPF external type 1 metric. This metric uses both internal and external costs when calculating the cost to access an external network.

type-2
OSPF external type 2 metric. This metric uses only external cost when calculating the cost to access an external network.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                metric-type
                     type-1
                     type-2
            }
        }
    }
```

```
}
```

The metric OSPF calculates the cost of accessing an external network.

Use the `set` form of this command to specify the OSPF external metric type for a route.

Use the `delete` form of this command to delete the metric type.

Use the `show` form of this command to display the metric type.

policy route route-map <map-name> rule <rule-num> set prepend-as { last-as <as-count> | own-as <as-count> }

Prepends the last-as, that is, the previous ASN or the own-as, that is, the user's ASN to the as-path of a route.

Syntax:

```
set policy route route-map map-name rule rule-num set prepend-as { last-as as-count | own-as as-count }
```

Syntax:

```
delete policy route route-map map-name rule rule-num set prepend-as [ last-as | own-as ]
```

Syntax:

```
show policy route route-map map-name rule rule-num set prepend-as
```

None

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

as-count
The number of times the last-as or own-as is prepended.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                prepend-as {
                   last-as as-count
                   own-as as-count
                }
            }
        }
    }
}
```

Use the `set` form of this command to prepend the last-as or the own-as to the existing as-path of a route. When all the match conditions in the route map rule are met, the last-as or own-as is prepended a specified number of times to the as-path of the route.

Use the `delete` form of this command to delete the prepend-as configuration from a route map rule.

Use the `show` form of this command to display the configuration for route maps.

Note: You can configure either the last-as or own-as option under a route map rule but not both.

policy route route-map <map-name> rule <rule-num> set origin

Modifies the BGP origin code of a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set origin`** { **`igp`** | **`egp`** | **`incomplete`** }

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set origin`** [**`igp`** | **`egp`** | **`incomplete`**]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
`egp`
Sets the BGP origin code to egp (Exterior Gateway Protocol).
`igp`
Sets the BGP origin code to igp (Interior Gateway Protocol).
`incomplete`
Sets the BGP origin code to incomplete.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                origin
                    igp
                    egp
                    incomplete
            }
        }
    }
}
```

Use the `set` form of this command to set the BGP origin code for packets that traverse a route map. When all the match conditions in the route map rule succeed, the BGP origin code is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set originator-id <ipv4>

Modifies the BGP originator ID attribute of a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set originator-id`** *ipv4*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set originator-id`** [*ipv4*]

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set originator-id`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
ipv4
The IPv4 address to be used as the new originator ID.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                originator-id ipv4
            }
        }
    }
}
```

Use the `set` form of this command to set the BGP originator ID for packets that traverse a route map. When all the match conditions in the route map rule succeed, the BGP originator ID is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set tag <tag>

Modifies the OSPF tag value of a route.

Syntax:
`set policy route route-map` *map-name* **`rule`** *rule-num* **`set tag`** *tag*

Syntax:
`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set tag`**

Syntax:
`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.
rule-num
The number of a defined route map rule.
tag
A 32-bit number representing the new value of the OSPF external Link-State Advertisement (LSA) tag field.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                tag tag
            }
        }
```

```
    }
}
```

Use the `set` form of this command to set the OSPF tag value for packets that traverse a route map. When all the match conditions in the route map rule succeed, the route tag is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

policy route route-map <map-name> rule <rule-num> set weight <weight>

Modifies the BGP weight of a route.

Syntax:

`set policy route route-map` *map-name* **`rule`** *rule-num* **`set weight`** *weight*

Syntax:

`delete policy route route-map` *map-name* **`rule`** *rule-num* **`set weight`**

Syntax:

`show policy route route-map` *map-name* **`rule`** *rule-num* **`set`**

map-name
The name of a defined route map.

rule-num
The number of a defined route map rule.

weight
The BGP weight to be recorded in the routing table. The range is 0 to 65535.

Configuration mode

```
policy {
    route-map map-name {
        rule rule-num {
            set {
                weight weight
            }
        }
    }
}
```

Use the `set` form of this command to set the BGP weight for routes. When all the match conditions in the route map rule succeed, the route weight is modified as specified.

Use the `delete` form of this command to delete this statement from the route map rule.

Use the `show` form of this command to display **`set`** statement configuration for route maps.

show ip access-list

Displays all IP access lists.

Syntax:

```
show ip access-list
```

Operational mode

Use this command to display IP access lists.

The following example shows IP access lists.

```
vyatta@vyatta:~$show ip access-list
ZEBRA:
Standard IP access list 1
    permit any
RIP:
Standard IP access list 1
    permit any
OSPF:
Standard IP access list 1
    permit any
BGP:
Standard IP access list 1
    permit any
```

show ip as-path-access-list

Displays all AS-path access lists.

Syntax:

```
show ip as-path-access-list
```

Operational mode

Use this command to display AS-path access lists.

The following example shows AS-path access lists.

```
vyatta@vyatta:~$ show ip as-path-access-list
AS path access list IN
    permit 50:1
vyatta@vyatta:~$
```

show ip community-list

Displays all IP community lists.

Syntax:

```
show ip community-list
```

Operational mode

Use this command to display community lists.

The following example shows community lists.

```
vyatta@vyatta:~$ show ip community-list
Community (expanded) access list 101
    permit AB*
vyatta@vyatta:~$
```

show ip extcommunity-list

Displays all extended IP community lists.

Syntax:

```
show ip extcommunity-list
```

Operational mode

Use this command to display extended IP community lists.

The following example shows extended IP community lists.

```
vyatta@vyatta:~$ show ip extcommunity-list
Community (expanded) access list 101
    permit AB*
vyatta@vyatta:~$
```

show ip prefix-list

Displays IP prefix lists.

Syntax:

```
show ip prefix-list [ detail | summary | list-name [ seq seq-num | ipv4net [ first-match | longer ] ] ]
```

detail
Displays detailed information for all IP prefix lists.

summary
Displays summary information for all IP prefix lists.

list-name
Displays information about the named IP prefix list.

seq-num
Displays the specified sequence from the named IP prefix list.

ipv4net
Displays the select prefix of the named IP prefix list.

first-match
Displays the first match from the select prefix of the named IP prefix list.

longer
Displays the longer match of the select prefix from the named IP prefix list.

Operational mode

Use this command to display prefix lists.

The following example shows prefix lists.

```
vyatta@vyatta:~$ show ip prefix-list
ZEBRA: ip prefix-list ABC: 1 entries
   seq 1 permit 192.168.2.0/24 ge 25
RIP: ip prefix-list ABC: 1 entries
   seq 1 permit 192.168.2.0/24 ge 25
OSPF: ip prefix-list ABC: 1 entries
   seq 1 permit 192.168.2.0/24 ge 25
BGP: ip prefix-list ABC: 1 entries
   seq 1 permit 192.168.2.0/24 ge 25
vyatta@vyatta:~$
```

show ip protocol

Displays IP route maps per protocol.

Syntax:

```
show ip protocol
```

Operational mode

Use this command to display IP route maps per protocol.

The following example shows IP route maps by protocol.

```
vyatta@vyatta:~$ show ip protocol
Protocol    : route-map
------------------------
system      : none
kernel      : none
connected   : none
static      : none
rip         : none
ripng       : none
ospf        : none
ospf6       : none
isis        : none
bgp         : none
hsls        : none
any         : none
vyatta@vyatta:~$
```

show route-map

Displays route map information.

Syntax:

```
show route-map [ map-name ]
```

map-name

The name for the route map.

Operational mode

Use this command to display route map information.

The following example shows route map information.

```
vyatta@vyatta:~$ show route-map
route-map rt1, permit, sequence 10
  Match clauses:
    ip address prefix-list: p1
  Set clauses:
```

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access