

VYATTA, INC.



Vyatta System

# Routing Policies

REFERENCE GUIDE

Routing Policies



Vyatta  
Suite 200  
1301 Shoreway Road  
Belmont, CA 94002  
vyatta.com  
650 413 7200  
1 888 VYATTA 1 (US and Canada)

## **COPYRIGHT**

Copyright © 2005–2012 Vyatta, Inc. All rights reserved.

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at [vyatta.com](http://vyatta.com).

## **PROPRIETARY NOTICES**

Vyatta is a registered trademark of Vyatta, Inc.

Hyper-V is a registered trademark of Microsoft Corporation.

VMware, VMware ESX, and VMware server are trademarks of VMware, Inc.

XenServer, and XenCenter are trademarks of Citrix Systems, Inc.

All other trademarks are the property of their respective owners.

RELEASE DATE: October 2012

DOCUMENT REVISION. 6.5R1 v01

RELEASED WITH: 6.5R1

PART NO. A0-0232-10-0013

# Contents

<b>Quick List of Commands</b> .....	<b>vii</b>
<b>List of Examples</b> .....	<b>x</b>
<b>Preface</b> .....	<b>xi</b>
Intended Audience .....	xii
Organization of This Guide .....	xii
Document Conventions .....	xiii
Vyatta Publications .....	xiii
<b>Chapter 1 Routing Policy Overview</b> .....	<b>1</b>
Routing Policy .....	2
<b>Chapter 2 Routing Policy Configuration Examples</b> .....	<b>3</b>
Filtering Routes using Access Lists .....	4
Basic RIP Configuration .....	4
Verifying the RIP Configuration .....	5
R3: show ip route .....	5
R3: show ip rip .....	6
Creating a Route Filtering Policy .....	7
Applying a Route Filtering Policy .....	8
Verifying the Route Filtering Policy Configuration .....	9
R3: show ip route .....	9
R3: show ip rip .....	10
Filtering Inbound Routes using Prefix Lists .....	11
Prefix List Configuration .....	11
Verifying the Inbound Filter .....	16
R1: show ip bgp .....	16
R1: show ip bgp .....	16
R4: show ip bgp .....	17
R4: show ip bgp .....	18
Filtering Outbound Routes using AS Path Lists .....	18
AS-path-list Configuration .....	18
Verifying the Outbound Filter .....	22
AS 200: show ip bgp .....	23
AS 200: show ip bgp .....	23

<b>Chapter 3 Routing Policy Commands</b> . . . . .	<b>25</b>
policy access-list <list-num> . . . . .	31
policy access-list <list-num> description <desc> . . . . .	32
policy access-list <list-num> rule <rule-num> . . . . .	33
policy access-list <list-num> rule <rule-num> action . . . . .	34
policy access-list <list-num> rule <rule-num> description <desc> . . . . .	36
policy access-list <list-num> rule <rule-num> destination . . . . .	38
policy access-list <list-num> rule <rule-num> source . . . . .	40
policy access-list6 <list-name> . . . . .	42
policy access-list6 <list-name> description <desc> . . . . .	43
policy access-list6 <list-name> rule <rule-num> . . . . .	44
policy access-list6 <list-name> rule <rule-num> action . . . . .	45
policy access-list6 <list-name> rule <rule-num> description <desc> . . . . .	47
policy access-list6 <list-name> rule <rule-num> destination . . . . .	49
policy access-list6 <list-name> rule <rule-num> source . . . . .	51
policy as-path-list <list-name> . . . . .	53
policy as-path-list <list-name> description <desc> . . . . .	54
policy as-path-list <list-name> rule <rule-num> . . . . .	55
policy as-path-list <list-name> rule <rule-num> action . . . . .	56
policy as-path-list <list-name> rule <rule-num> description <desc> . . . . .	58
policy as-path-list <list-name> rule <rule-num> regex <regex> . . . . .	60
policy community-list <list-num> . . . . .	62
policy community-list <list-num> description <desc> . . . . .	63
policy community-list <list-num> rule <rule-num> . . . . .	64
policy community-list <list-num> rule <rule-num> action . . . . .	65
policy community-list <list-num> rule <rule-num> description <desc> . . . . .	67
policy community-list <list-num> rule <rule-num> regex <regex> . . . . .	69
policy prefix-list <list-name> . . . . .	71
policy prefix-list <list-name> description <desc> . . . . .	72
policy prefix-list <list-name> rule <rule-num> . . . . .	73
policy prefix-list <list-name> rule <rule-num> action . . . . .	74
policy prefix-list <list-name> rule <rule-num> description <desc> . . . . .	76
policy prefix-list <list-name> rule <rule-num> ge <value> . . . . .	78
policy prefix-list <list-name> rule <rule-num> le <value> . . . . .	80
policy prefix-list <list-name> rule <rule-num> prefix <ipv4net> . . . . .	82
policy prefix-list6 <list-name> . . . . .	84
policy prefix-list6 <list-name> description <desc> . . . . .	85
policy prefix-list6 <list-name> rule <rule-num> . . . . .	86
policy prefix-list6 <list-name> rule <rule-num> action . . . . .	87
policy prefix-list6 <list-name> rule <rule-num> description <desc> . . . . .	89
policy prefix-list6 <list-name> rule <rule-num> ge <value> . . . . .	91

policy prefix-list6 <list-name> rule <rule-num> le <value> . . . . .	93
policy prefix-list6 <list-name> rule <rule-num> prefix <ipv6net> . . . . .	95
policy route-map <map-name> . . . . .	97
policy route-map <map-name> description <desc> . . . . .	98
policy route-map <map-name> rule <rule-num> . . . . .	99
policy route-map <map-name> rule <rule-num> action . . . . .	100
policy route-map <map-name> rule <rule-num> call <target> . . . . .	102
policy route-map <map-name> rule <rule-num> continue <target-num> . . . . .	104
policy route-map <map-name> rule <rule-num> description <desc> . . . . .	106
policy route-map <map-name> rule <rule-num> match as-path <list-name> . . . . .	108
policy route-map <map-name> rule <rule-num> match community . . . . .	110
policy route-map <map-name> rule <rule-num> match interface <ethx> . . . . .	112
policy route-map <map-name> rule <rule-num> match ip address . . . . .	114
policy route-map <map-name> rule <rule-num> match ip nexthop . . . . .	116
policy route-map <map-name> rule <rule-num> match ip route-source . . . . .	118
policy route-map <map-name> rule <rule-num> match ipv6 address . . . . .	120
policy route-map <map-name> rule <rule-num> match ipv6 nexthop . . . . .	122
policy route-map <map-name> rule <rule-num> match metric <metric> . . . . .	124
policy route-map <map-name> rule <rule-num> match origin . . . . .	126
policy route-map <map-name> rule <rule-num> match peer <ipv4> . . . . .	128
policy route-map <map-name> rule <rule-num> match tag <tag> . . . . .	130
policy route-map <map-name> rule <rule-num> on-match . . . . .	132
policy route-map <map-name> rule <rule-num> set aggregator . . . . .	134
policy route-map <map-name> rule <rule-num> set as-path-prepend <prepend> . . . . .	136
policy route-map <map-name> rule <rule-num> set atomic-aggregate . . . . .	138
policy route-map <map-name> rule <rule-num> set comm-list . . . . .	140
policy route-map <map-name> rule <rule-num> set community . . . . .	142
policy route-map <map-name> rule <rule-num> set ip-next-hop <ipv4> . . . . .	144
policy route-map <map-name> rule <rule-num> set ipv6-next-hop <scope> <ipv6> . . . . .	146
policy route-map <map-name> rule <rule-num> set local-preference <local-pref> . . . . .	148
policy route-map <map-name> rule <rule-num> set metric <metric> . . . . .	150
policy route-map <map-name> rule <rule-num> set metric-type <type> . . . . .	152
policy route-map <map-name> rule <rule-num> set origin . . . . .	154
policy route-map <map-name> rule <rule-num> set originator-id <ipv4> . . . . .	156
policy route-map <map-name> rule <rule-num> set tag <tag> . . . . .	158
policy route-map <map-name> rule <rule-num> set weight <weight> . . . . .	160
show ip access-list . . . . .	162
show ip as-path-access-list . . . . .	164
show ip community-list . . . . .	165
show ip extcommunity-list . . . . .	166
show ip prefix-list . . . . .	167

show ip protocol ..... 169

show route-map..... 171

**Glossary of Acronyms ..... 173**

# Quick List of Commands

Use this list to help you quickly locate commands.

policy access-list <list-num> description <desc> . . . . .	32
policy access-list <list-num> rule <rule-num> action . . . . .	34
policy access-list <list-num> rule <rule-num> description <desc> . . . . .	36
policy access-list <list-num> rule <rule-num> destination. . . . .	38
policy access-list <list-num> rule <rule-num> source . . . . .	40
policy access-list <list-num> rule <rule-num>. . . . .	33
policy access-list <list-num> . . . . .	31
policy access-list6 <list-name> description <desc> . . . . .	43
policy access-list6 <list-name> rule <rule-num> action . . . . .	45
policy access-list6 <list-name> rule <rule-num> description <desc> . . . . .	47
policy access-list6 <list-name> rule <rule-num> destination. . . . .	49
policy access-list6 <list-name> rule <rule-num> source . . . . .	51
policy access-list6 <list-name> rule <rule-num>. . . . .	44
policy access-list6 <list-name> . . . . .	42
policy as-path-list <list-name> description <desc>. . . . .	54
policy as-path-list <list-name> rule <rule-num> action . . . . .	56
policy as-path-list <list-name> rule <rule-num> description <desc> . . . . .	58
policy as-path-list <list-name> rule <rule-num> regex <regex> . . . . .	60
policy as-path-list <list-name> rule <rule-num>. . . . .	55
policy as-path-list <list-name> . . . . .	53
policy community-list <list-num> description <desc> . . . . .	63
policy community-list <list-num> rule <rule-num> action. . . . .	65
policy community-list <list-num> rule <rule-num> description <desc> . . . . .	67
policy community-list <list-num> rule <rule-num> regex <regex> . . . . .	69
policy community-list <list-num> rule <rule-num> . . . . .	64
policy community-list <list-num>. . . . .	62
policy prefix-list <list-name> description <desc> . . . . .	72
policy prefix-list <list-name> rule <rule-num> action . . . . .	74
policy prefix-list <list-name> rule <rule-num> description <desc> . . . . .	76
policy prefix-list <list-name> rule <rule-num> ge <value>. . . . .	78
policy prefix-list <list-name> rule <rule-num> le <value> . . . . .	80
policy prefix-list <list-name> rule <rule-num> prefix <ipv4net> . . . . .	82

policy prefix-list <list-name> rule <rule-num> .....	73
policy prefix-list <list-name> .....	71
policy prefix-list6 <list-name> description <desc> .....	85
policy prefix-list6 <list-name> rule <rule-num> action .....	87
policy prefix-list6 <list-name> rule <rule-num> description <desc>.....	89
policy prefix-list6 <list-name> rule <rule-num> ge <value>.....	91
policy prefix-list6 <list-name> rule <rule-num> le <value> .....	93
policy prefix-list6 <list-name> rule <rule-num> prefix <ipv6net>.....	95
policy prefix-list6 <list-name> rule <rule-num> .....	86
policy prefix-list6 <list-name> .....	84
policy route-map <map-name> description <desc>.....	98
policy route-map <map-name> rule <rule-num> action .....	100
policy route-map <map-name> rule <rule-num> call <target>.....	102
policy route-map <map-name> rule <rule-num> continue <target-num>.....	104
policy route-map <map-name> rule <rule-num> description <desc> .....	106
policy route-map <map-name> rule <rule-num> match as-path <list-name>.....	108
policy route-map <map-name> rule <rule-num> match community .....	110
policy route-map <map-name> rule <rule-num> match interface <ethx>.....	112
policy route-map <map-name> rule <rule-num> match ip address .....	114
policy route-map <map-name> rule <rule-num> match ip nexthop .....	116
policy route-map <map-name> rule <rule-num> match ip route-source.....	118
policy route-map <map-name> rule <rule-num> match ipv6 address .....	120
policy route-map <map-name> rule <rule-num> match ipv6 nexthop .....	122
policy route-map <map-name> rule <rule-num> match metric <metric> .....	124
policy route-map <map-name> rule <rule-num> match origin.....	126
policy route-map <map-name> rule <rule-num> match peer <ipv4> .....	128
policy route-map <map-name> rule <rule-num> match tag <tag>.....	130
policy route-map <map-name> rule <rule-num> on-match .....	132
policy route-map <map-name> rule <rule-num> set aggregator .....	134
policy route-map <map-name> rule <rule-num> set as-path-prepend <prepend> .....	136
policy route-map <map-name> rule <rule-num> set atomic-aggregate .....	138
policy route-map <map-name> rule <rule-num> set comm-list .....	140
policy route-map <map-name> rule <rule-num> set community .....	142
policy route-map <map-name> rule <rule-num> set ip-next-hop <ipv4> .....	144
policy route-map <map-name> rule <rule-num> set ipv6-next-hop <scope> <ipv6>.....	146
policy route-map <map-name> rule <rule-num> set local-preference <local-pref>.....	148



---

policy route-map <map-name> rule <rule-num> set metric <metric> . . . . .	150
policy route-map <map-name> rule <rule-num> set metric-type <type> . . . . .	152
policy route-map <map-name> rule <rule-num> set origin . . . . .	154
policy route-map <map-name> rule <rule-num> set originator-id <ipv4>. . . . .	156
policy route-map <map-name> rule <rule-num> set tag <tag> . . . . .	158
policy route-map <map-name> rule <rule-num> set weight <weight> . . . . .	160
policy route-map <map-name> rule <rule-num> . . . . .	99
policy route-map <map-name> . . . . .	97
show ip access-list . . . . .	162
show ip as-path-access-list . . . . .	164
show ip community-list . . . . .	165
show ip extcommunity-list . . . . .	166
show ip prefix-list . . . . .	167
show ip protocol . . . . .	169
show route-map . . . . .	171

# List of Examples

Use this list to help you locate examples you'd like to look at or try.

Example 3-1 “show ip access-list”: Displaying IP access lists .....	162
Example 3-2 “show ip as-path-access-list”: Displaying as-path access lists .....	164
Example 3-3 “show ip community-list”: Displaying community lists .....	165
Example 3-4 “show ip extcommunity-list”: Displaying extended IP community lists .....	166
Example 3-5 “show ip prefix-list”: Displaying prefix lists .....	168
Example 3-6 “show ip protocol”: Displaying IP route maps by protocol .....	169
Example 3-7 “show route-map”: Displaying route map information .....	171

# Preface

This document describes the various deployment, installation, and upgrade options for Vyatta software.

This preface provides information about using this guide. The following topics are presented:

- [Intended Audience](#)
- [Organization of This Guide](#)
- [Document Conventions](#)
- [Vyatta Publications](#)

## Intended Audience

---

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP services

## Organization of This Guide

---

This guide has the following aid to help you find the information you are looking for:

- [Quick List of Commands](#)  
Use this list to help you quickly locate commands.
- [List of Examples](#)  
Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
<a href="#">Chapter 1: Routing Policy Overview</a>	This chapter provides a brief overview of routing policy features on the Vyatta system.	1
<a href="#">Chapter 2: Routing Policy Configuration Examples</a>	This chapter provides configuration examples for routing policies.	3
<a href="#">Chapter 3: Routing Policy Commands</a>	This chapter describes Vyatta system routing policy commands.	25
<a href="#">Glossary of Acronyms</a>		173

# Document Conventions

This guide uses the following advisory paragraphs, as follows.



**WARNING** Warnings alert you to situations that may pose a threat to personal safety.



**CAUTION** Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.

**NOTE** Notes provide information you might need to avoid problems or configuration errors.

This document uses the following typographic conventions.

Monospace	Examples, command-line output, and representations of configuration nodes.
<b>bold Monospace</b>	Your input: something you type at a command line.
<b>bold</b>	Commands, keywords, and file names, when mentioned inline.  Objects in the user interface, such as tabs, buttons, screens, and panes.
<i>italics</i>	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[ key1   key2]	Enumerated options for completing a syntax. An example is [enable   disable].
<i>num1–numN</i>	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
<i>arg1..argN</i>	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
<i>arg[ arg...]</i> <i>arg[,arg...]</i>	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

## Vyatta Publications

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Additional information is available on [www.vyatta.com](http://www.vyatta.com) and [www.vyatta.org](http://www.vyatta.org).

# Chapter 1: Routing Policy Overview

This chapter provides a brief overview of routing policy features on the Vyatta system.

This chapter presents the following topics:

- [Routing Policy](#)

## 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 Vyatta 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* via 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 Vyatta system will be 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 Vyatta system 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 via 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.



# Chapter 2: Routing Policy Configuration Examples

This chapter provides configuration examples for routing policies.

This chapter presents the following topics:

- [Filtering Routes using Access Lists](#)
- [Filtering Inbound Routes using Prefix Lists](#)
- [Filtering Outbound Routes using AS Path Lists](#)

## Filtering Routes using Access Lists

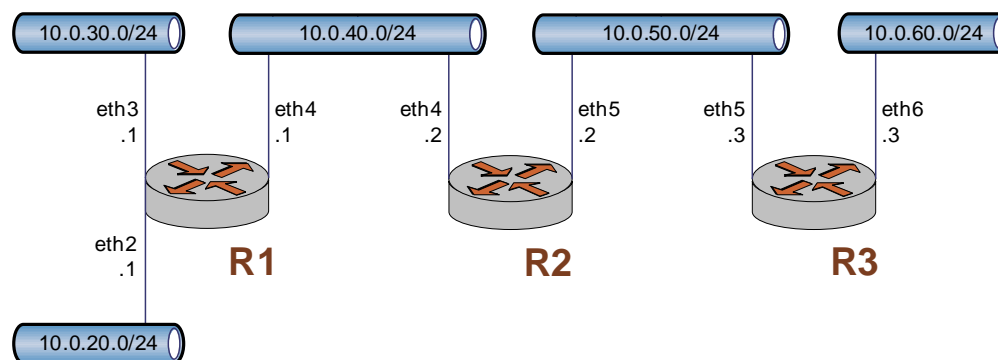
This section presents the following topics:

- [Basic RIP Configuration](#)
- [Verifying the RIP Configuration](#)
- [Creating a Route Filtering Policy](#)
- [Applying a Route Filtering Policy](#)
- [Verifying the Route Filtering Policy Configuration](#)

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 reference diagram in [Figure 2-1](#).

Figure 2-1 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.

Example 2-1 Basic RIP configuration

Router	Step	Command(s)
--------	------	------------

---

**Example 2-1 Basic RIP configuration**

---

R1	Display the configuration.	<pre>vyatta@R1# show protocols rip {   network 10.0.40.0/24   redistribute {     connected {     }   } }</pre>
----	----------------------------	--

---

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

---

R3	Display the configuration.	<pre>vyatta@R3# show protocols rip {   network 10.0.50.0/24   redistribute {     connected {     }   } }</pre>
----	----------------------------	--

---

## Verifying the RIP Configuration

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

### R3: show ip route

[Example 2-2](#) shows the output of the **show ip route** command for router R3.

---

**Example 2-2 Verifying RIP on R3: "show ip route"**

---

```
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, eth5, 00:20:16
R>* 10.0.30.0/24 [120/3] via 10.0.50.2, eth5, 00:34:04
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, eth5, 02:15:26
C>* 10.0.50.0/24 is directly connected, eth5
C>* 10.0.60.0/24 is directly connected, eth6
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 eth5 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 [Example 2-3](#).

### Example 2-3 Verifying RIP on R3: “show ip rip”

```

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.

### Example 2-4 Route filtering configuration

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

## Example 2-4 Route filtering configuration

R2	Display the configuration.	<pre> 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     }   } } </pre>
----	----------------------------	--

## Applying a Route Filtering Policy

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

## Example 2-5 Applying a route filtering policy

Router	Step	Command(s)
R2	Use the access list created in the previous example to filter incoming route advertisements.	<pre> vyatta@R2# set protocols rip distribute-list access-list in 100 </pre>
R2	Commit the configuration.	<pre> vyatta@R2# commit </pre>

---

**Example 2-5** Applying a route filtering policy

---

R2	Display the configuration.	<pre>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 {     }   } }</pre>
----	----------------------------	--

---

## 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

[Example 2-6](#) shows the output of the **show ip route** command for router R3.

---

**Example 2-6** Verifying routing policy changes on R3: "show ip route"

---

```
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, eth5, 00:45:21
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, eth5, 00:45:21
C>* 10.0.50.0/24 is directly connected, eth5
C>* 10.0.60.0/24 is directly connected, eth6
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 eth5 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 [Example 2-7](#).

Example 2-7 Verifying routing policy changes on R3: “show ip rip”

---

```
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 Vyatta system 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.

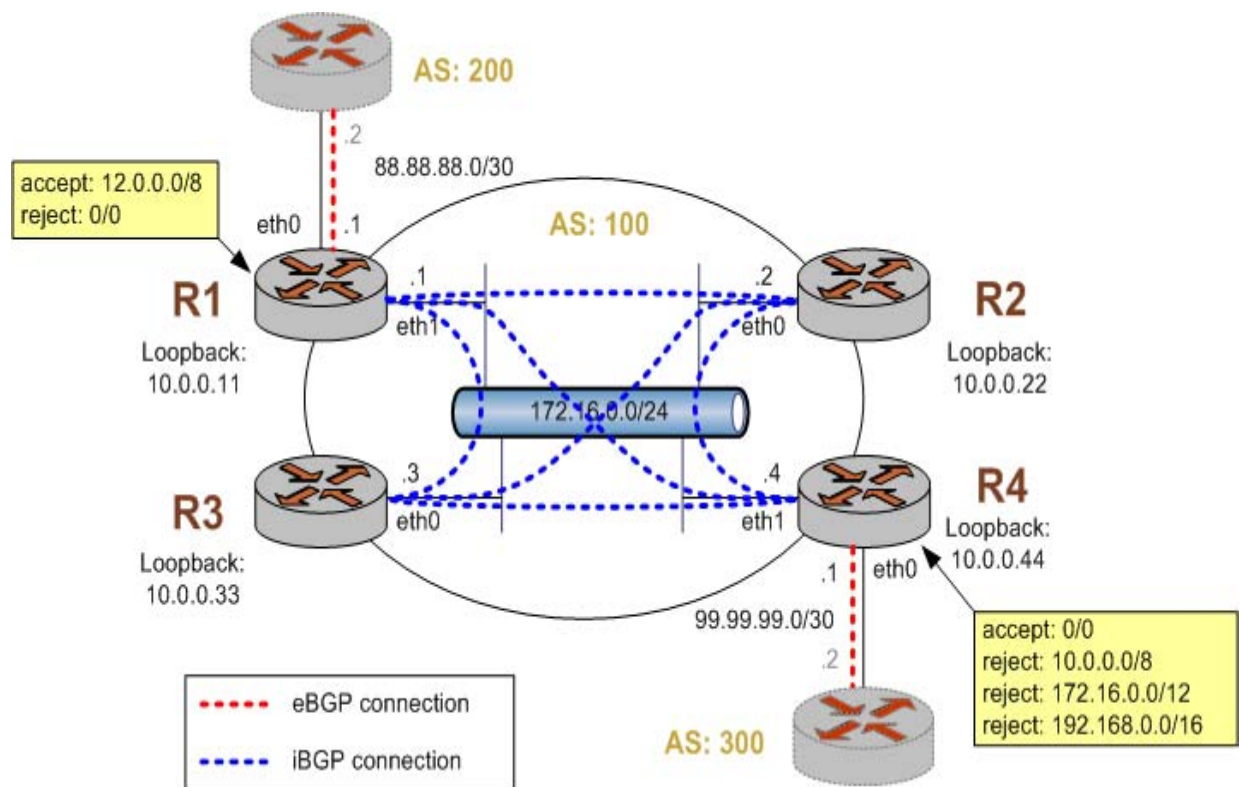
[Example 2-8](#) creates the following inbound filtering policies:

- 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 [Figure 2-2](#).

**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 2-2 Filtering inbound routes



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

#### Example 2-8 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.	<pre>vyatta@R1# set policy prefix-list ALLOW-PREFIXES rule 1 action permit vyatta@R1# set policy prefix-list ALLOW-PREFIXES rule 1 prefix 12.0.0.0/8</pre>
R1	Create a route map rule to permit all prefixes in our list.	<pre>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</pre>
R1	Create a route map rule to deny all other prefixes.	<pre>vyatta@R1# set policy route-map eBGP-IMPORT rule 20 action deny</pre>

## Example 2-8 Creating an import policy

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 route-map import eBGP-IMPORT
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 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 remote-as 200 route-map { import eBGP-IMPORT } vyatta@R1#

## Example 2-8 Creating an import policy

R4	Create a rule to match any prefix from 10.0.0.0/8 to 32.	<pre>vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 1 action permit vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 1 le 32 vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 1 prefix 10.0.0.0/8</pre>
R4	Create a rule to match any prefix from 172.16.0.0/12 to 32.	<pre>vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 2 action permit vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 2 le 32 vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 2 prefix 172.16.0.0/12</pre>
R4	Create a rule to match any prefix from 192.168.0.0/16 to 32.	<pre>vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 3 action permit vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 3 le 32 vyatta@R4# set policy prefix-list RFC1918PREFIXES rule 3 prefix 192.168.0.0/16</pre>
R4	Create a route map rule to deny all prefixes in our list.	<pre>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</pre>
R4	Create a route map rule to permit all other prefixes.	<pre>vyatta@R4# set policy route-map eBGP-IMPORT rule 20 action permit</pre>
R4	Assign the route map policy created as the import route map policy for AS 300.	<pre>vyatta@R4# set protocols bgp 100 neighbor 99.99.99.2 route-map import eBGP-IMPORT</pre>
R4	Commit the configuration.	<pre>vyatta@R4# commit</pre>
R4	Reset the BGP session to the peer so that the new policies are enabled.	<pre>vyatta@R1# run reset ip bgp 99.99.99.2</pre>

---

**Example 2-8** Creating an import policy

---

R4	Display the policy configuration.	<pre>vyatta@R4# show policy prefix-list RFC1918PREFIXES {     rule 1 {         action permit         le 32         prefix 10.0.0.0/8     }     rule 2 {         action permit         le 32         prefix 172.16.0.0/12     }     rule 3 {         action permit         le 32         prefix 192.168.0.0/16     } } route-map eBGP-IMPORT {     rule 10 {         action deny         match {             ip {                 address {                     prefix-list RFC1918PREFIXES                 }             }         }     }     rule 20 {         action permit     } } vyatta@R4#</pre>
R4	Display the BGP configuration for eBGP neighbor 99.99.99.2.	<pre>vyatta@R4# show protocols bgp 100 neighbor 99.99.99.2 remote-as 300 route-map {     import eBGP-IMPORT } vyatta@R4#</pre>

---

## Verifying the Inbound Filter

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

### R1: show ip bgp

[Example 2-9](#) shows R1's BGP table *before* the import filter is applied.

Example 2-9 R1 inbound BGP routes before import filtering

```
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

[Example 2-10](#) shows R1's BGP table *after* the import filter is applied.

Example 2-10 R1 inbound BGP routes after import filtering

```
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:~$

```

Note that only 12.0.0.0 from 88.88.88.2 is still in the table.

## R4: show ip bgp

Example 2-11 shows R4's BGP table *before* the import filter is applied.

Example 2-11 R4 inbound BGP routes before import filtering

```

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

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

Example 2-12 R4 inbound BGP routes after import filtering

```
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

Total number of prefixes 7
vyatta@R4:~$
```

## 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 Vyatta system 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 AS 200 or AS 300. This means that:

- eBGP routes from R1's eBGP peer (AS 200) should not be sent to R4's eBGP peer.



- Routes from R4's eBGP peer (AS 300) should not be sent to R1's eBGP peer.

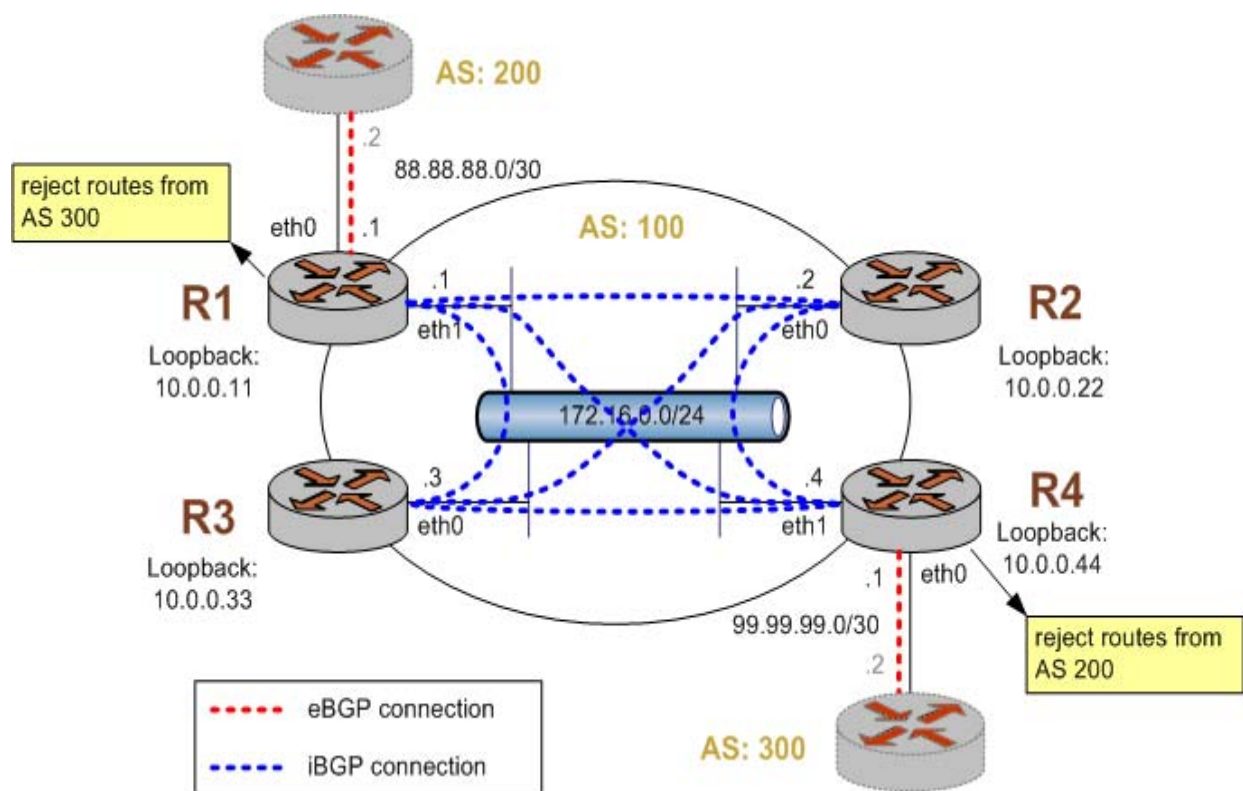
If we *did not* implement this filtering, AS 300 might send traffic destined for AS 200 to router R4, and this traffic would then be carried across the AS 100 network.

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 AS 100 from acting as a transit network for AS 200 and AS 300.

This export policy is shown in [Figure 2-3](#).

**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 2-3 Filtering outbound routes



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

#### Example 2-13 Creating an export policy

Router	Step	Command(s)
--------	------	------------

---

**Example 2-13** Creating an export policy

---

R1	Create a list of AS paths to deny. In this case we just have one - AS300.	<pre>vyatta@R1# set policy as-path-list AS300 rule 1 action permit vyatta@R1# set policy as-path-list AS300 rule 1 regex 300</pre>
R1	Create a route map rule to deny all AS paths in our list.	<pre>vyatta@R1# set policy route-map eBGP-EXPORT rule 10 action deny vyatta@R1# set policy route-map eBGP-EXPORT rule 10 match as-path AS300</pre>
R1	Create a route map rule to permit all other prefixes.	<pre>vyatta@R1# set policy route-map eBGP-EXPORT rule 20 action permit</pre>
R1	Assign the route map policy created as the export route map policy for AS 200.	<pre>vyatta@R1# set protocols bgp 100 neighbor 88.88.88.2 route-map export eBGP-EXPORT</pre>
R1	Commit the configuration.	<pre>vyatta@R1# commit</pre>
R1	Reset the BGP session to the peer so that the new policies are enabled.	<pre>vyatta@R1# run reset ip bgp 88.88.88.2</pre>

---

## Example 2-13 Creating an export policy

R1	Display the policy configurations	<pre>vyatta@R1# show policy as-path-list AS300 rule 1 {     action permit     regex 300 } vyatta@R1# show policy route-map eBGP-EXPORT rule 10 {     action deny     match {         as-path AS300     } } rule 20 {     action permit } vyatta@R1#</pre>
R1	Display the BGP configuration for eBGP neighbor 88.88.88.2.	<pre>vyatta@R1# show protocols bgp 100 neighbor 88.88.88.2 remote-as 200 route-map {     export eBGP-EXPORT     import eBGP-IMPORT } vyatta@R1#</pre>
R4	Create a list of AS paths to deny. In this case we just have one - AS200.	<pre>vyatta@R4# set policy as-path-list AS200 rule 1 action permit vyatta@R4# set policy as-path-list AS200 rule 1 regex 200</pre>
R4	Create a route map rule to deny all AS paths in our list.	<pre>vyatta@R4# set policy route-map eBGP-EXPORT rule 10 action deny vyatta@R4# set policy route-map eBGP-EXPORT rule 10 match as-path AS200</pre>
R4	Create a route map rule to permit all other prefixes.	<pre>vyatta@R4# set policy route-map eBGP-EXPORT rule 20 action permit</pre>

## Example 2-13 Creating an export policy

R4	Assign the route map policy created as the export route map policy for AS 300.	vyatta@R4# set protocols bgp 100 neighbor 99.99.99.2 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
R4	Display the policy configurations.	vyatta@R4# show policy as-path-list AS200 rule 1 { action permit regex 200 } vyatta@R4# show policy route-map eBGP-EXPORT rule 10 { action deny match { as-path AS200 } } 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 remote-as 300 route-map { export eBGP-EXPORT import eBGP-IMPORT } vyatta@R4#

## Verifying the Outbound Filter

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

## AS 200: show ip bgp

Example 2-14 shows AS 200's BGP table *before* the export filter is applied.

Example 2-14 AS 200 outbound BGP routes before export filtering

```
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

Example 2-15 shows AS 200's BGP table *after* the export filter is applied.

Example 2-15 AS 200 outbound BGP routes after export filtering

```
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:~\$

# Chapter 3: Routing Policy Commands

This chapter describes Vyatta system routing policy commands.

This chapter contains the following commands.

<b>Configuration Commands</b>	
<b>Access Lists</b>	
<code>policy access-list &lt;list-num&gt;</code>	Defines an access list.
<code>policy access-list &lt;list-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an access list.
<code>policy access-list &lt;list-num&gt; rule &lt;rule-num&gt;</code>	Creates a rule for an access list.
<code>policy access-list &lt;list-num&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching an access list rule.
<code>policy access-list &lt;list-num&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an access list rule.
<code>policy access-list &lt;list-num&gt; rule &lt;rule-num&gt; destination</code>	Defines match criteria for an access list rule based on destination.
<code>policy access-list &lt;list-num&gt; rule &lt;rule-num&gt; source</code>	Defines match criteria for an access list rule based on source.
<b>IPv6 Access Lists</b>	
<code>policy access-list6 &lt;list-name&gt;</code>	Defines an IPv6 access list.
<code>policy access-list6 &lt;list-name&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an IPv6 access list.
<code>policy access-list6 &lt;list-name&gt; rule &lt;rule-num&gt;</code>	Creates a rule for an IPv6 access list.
<code>policy access-list6 &lt;list-name&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching an IPv6 access list rule.
<code>policy access-list6 &lt;list-name&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an IPv6 access list rule.
<code>policy access-list6 &lt;list-name&gt; rule &lt;rule-num&gt; source</code>	Defines match criteria for an IPv6 access list rule based on source.
<b>AS Path Lists</b>	
<code>policy as-path-list &lt;list-name&gt;</code>	Defines an autonomous system (AS) path list.
<code>policy as-path-list &lt;list-name&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an AS path list.
<code>policy as-path-list &lt;list-name&gt; rule &lt;rule-num&gt;</code>	Creates a rule for an AS path list.



<code>policy as-path-list &lt;list-name&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching an AS path list rule.
<code>policy as-path-list &lt;list-name&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an AS path list rule.
<code>policy as-path-list &lt;list-name&gt; rule &lt;rule-num&gt; regex &lt;regex&gt;</code>	Defines match criteria for an AS path list rule based on a regular expression.
<b>Community Lists</b>	
<code>policy community-list &lt;list-num&gt;</code>	Defines a BGP community list.
<code>policy community-list &lt;list-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a community list.
<code>policy community-list &lt;list-num&gt; rule &lt;rule-num&gt;</code>	Creates a rule for a community list.
<code>policy community-list &lt;list-num&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching a community list rule.
<code>policy community-list &lt;list-num&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a community list rule.
<code>policy community-list &lt;list-num&gt; rule &lt;rule-num&gt; regex &lt;regex&gt;</code>	Defines match criteria for a community list rule based on a regular expression.
<b>Prefix Lists</b>	
<code>policy prefix-list &lt;list-name&gt;</code>	Defines a prefix list.
<code>policy prefix-list &lt;list-name&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a prefix list.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt;</code>	Creates a rule for a prefix list.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching a prefix list rule.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a prefix list rule.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt; ge &lt;value&gt;</code>	Defines match criteria for a prefix list rule based on a “greater-than-or-equal-to” numeric comparison.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt; le &lt;value&gt;</code>	Defines a match criterion based on a “less-than-or-equal-to” numeric comparison for a prefix list rule.
<code>policy prefix-list &lt;list-name&gt; rule &lt;rule-num&gt; prefix &lt;ipv4net&gt;</code>	Defines match criteria for a prefix list rule based on an IPv4 network.

IPv6 Prefix Lists	
<code>policy prefix-list6 &lt;list-name&gt;</code>	Defines an IPv6 prefix list.
<code>policy prefix-list6 &lt;list-name&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an IPv6 prefix list.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt;</code>	Creates a rule for an IPv6 prefix list.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching an IPv6 prefix list rule.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for an IPv6 prefix list rule.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt; ge &lt;value&gt;</code>	Defines match criteria for an IPv6 prefix list rule based on a “greater-than-or-equal-to” numeric comparison.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt; le &lt;value&gt;</code>	Defines a match criterion based on a “less-than-or-equal-to” numeric comparison for an IPv6 prefix list rule.
<code>policy prefix-list6 &lt;list-name&gt; rule &lt;rule-num&gt; prefix &lt;ipv6net&gt;</code>	Defines match criteria for a prefix list rule based on an IPv6 network.
Route Maps	
<code>policy route-map &lt;map-name&gt;</code>	Defines a route map for policy-based routing.
<code>policy route-map &lt;map-name&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a route map.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt;</code>	Creates a rule for a route map.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; action</code>	Specifies the action to be taken for packets matching a route map rule.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; call &lt;target&gt;</code>	Calls to another route map.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; continue &lt;target-num&gt;</code>	Calls to another rule within the current route map.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; description &lt;desc&gt;</code>	Allows you to specify a brief description for a route map rule.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match as-path &lt;list-name&gt;</code>	Defines a match condition for a route map based on an AS path list
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match community</code>	Defines a match condition for a route map based on BGP communities.

<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match interface &lt;ethx&gt;</code>	Defines a match condition for a route map based on the first-hop interface.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match ip address</code>	Defines a match condition for a route map based on IP address.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match ip nexthop</code>	Defines a match condition for a route map based on the next-hop address.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match ip route-source</code>	Defines a match condition for a route map based on the address from where a route is advertised.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match ipv6 address</code>	Defines a match condition for a route map based on IPv6 address.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match ipv6 nexthop</code>	Defines a match condition for a route map based on the next-hop IPv6 address.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match metric &lt;metric&gt;</code>	Defines a match condition for a route map based on the route's metric.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match origin</code>	Defines a match condition for a route map based on the route's origin.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match peer &lt;ipv4&gt;</code>	Defines a match condition for a route map based on peer IP address.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; match tag &lt;tag&gt;</code>	Defines a match condition for a route map based on OSPF tag.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; on-match</code>	Specifies an alternative exit policy for a route map.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set aggregator</code>	Modifies the BGP aggregator attribute of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set as-path-prepend &lt;prepend&gt;</code>	Sets or prepends to the AS path of the route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set atomic-aggregate</code>	Sets the BGP atomic-aggregate attribute in a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set comm-list</code>	Modifies the BGP community list in a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set community</code>	Modifies the BGP communities attribute in a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set ip-next-hop &lt;ipv4&gt;</code>	Modifies the next hop destination of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set ipv6-next-hop &lt;scope&gt; &lt;ipv6&gt;</code>	Modifies the IPv6 next hop destination of a route.

<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set local-preference &lt;local-pref&gt;</code>	Modifies the BGP local-pref attribute in a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set metric &lt;metric&gt;</code>	Modifies the metric of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set metric-type &lt;type&gt;</code>	Specifies the OSPF external metric-type for a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set origin</code>	Modifies the BGP origin code of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set originator-id &lt;ipv4&gt;</code>	Modifies the BGP originator ID attribute of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set tag &lt;tag&gt;</code>	Modifies the OSPF tag value of a route.
<code>policy route-map &lt;map-name&gt; rule &lt;rule-num&gt; set weight &lt;weight&gt;</code>	Modifies the BGP weight of a route.
<b>Operational Commands</b>	
<code>show ip access-list</code>	Displays all IP access lists.
<code>show ip as-path-access-list</code>	Displays all as-path access lists.
<code>show ip community-list</code>	Displays all IP community lists.
<code>show ip extcommunity-list</code>	Displays all extended IP community lists.
<code>show ip prefix-list</code>	Displays IP prefix lists.
<code>show ip protocol</code>	Displays IP route maps per protocol.
<code>show route-map</code>	Displays route map information.

## policy access-list <list-num>

Defines an access list.

---

### Syntax

```
set policy access-list list-num
delete policy access-list list-num
show policy access-list list-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-num</i>	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 <b>policy access-list</b> configuration nodes.
-----------------	---

---

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy access-list list-num description desc
delete policy access-list list-num description
show policy access-list list-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-num</i>	The number of a defined access list.
<i>desc</i>	A brief text description for the access list.

---

### Default

None.

---

### Usage Guidelines

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 access-list <list-num> rule <rule-num>

Creates a rule for an access list.

---

### Syntax

```
set policy access-list list-num rule rule-num
delete policy access-list list-num rule rule-num
show policy access-list list-num rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-num</i>	The number of a defined access list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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

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

### Syntax

```
set policy access-list list-num rule rule-num action {deny | permit}
delete policy access-list list-num rule rule-num action
show policy access-list list-num rule rule-num action
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>list-num</i>	The number of a defined access list.
<i>rule-num</i>	The number of a defined access list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

### Default

Packets matching this rule are forwarded.



---

### Usage Guidelines

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

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

---

### Syntax

```
set policy access-list list-num rule rule-num description desc
delete policy access-list list-num rule rule-num description
show policy access-list list-num rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-num</i>	The number of a defined access list.
<i>rule-num</i>	The number of a defined access list rule.
<i>desc</i>	A brief text description for the access list rule.

---

### Default

None.

---

### Usage Guidelines

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

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

### Syntax

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

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

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

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>list-num</i>	The number of a defined access list.
<i>rule-num</i>	The number of a defined access list.
<b>any</b>	Match packets destined for any destination. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>host <i>ipv4</i></b>	Match packets destined for the specified IPv4 host. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

<b>inverse-mask</b> <i>ipv4</i>	Match packets destined for the network specified by the mask. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>network</b> <i>ipv4net</i>	Match packets destined for the specified network. The format is <i>ip-address/prefix</i> . Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

---

### Default

None.

---

### Usage Guidelines

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

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

### Syntax

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

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

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

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>list-num</i>	The number of a defined access list.
<i>rule-num</i>	The number of a defined access list rule.
<b>any</b>	Match packets coming from any source. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>host <i>ipv4</i></b>	Match packets coming from the specified IPv4 host. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

<b>inverse-mask</b> <i>ipv4</i>	Match packets coming from the network specified by the mask. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>network</b> <i>ipv4net</i>	Match packets coming from the specified network. The format is <i>ip-address/prefix</i> . Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

---

### Default

None.

---

### Usage Guidelines

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 access-list6 <list-name>

Defines an IPv6 access list.

---

### Syntax

```
set policy access-list6 list-name
delete policy access-list6 list-name
show policy access-list6 list-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	Multi-node. The name of an IPv6 access list. You can create multiple access lists by creating multiple <b>policy access-list</b> configuration nodes.
------------------	--

---

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy access-list6 list-name description desc
delete policy access-list6 list-name description
show policy access-list6 list-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of an IPv6 access list.
<i>desc</i>	A brief text description for the access list.

---

### Default

None.

---

### Usage Guidelines

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

Creates a rule for an IPv6 access list.

---

### Syntax

```
set policy access-list6 list-name rule rule-num
delete policy access-list6 list-name rule rule-num
show policy access-list6 list-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	The name of an IPv6 access list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 65535.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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 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 access-list6 list-name rule rule-num action {deny | permit}
delete policy access-list6 list-name rule rule-num action
show policy access-list6 list-name rule rule-num action
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of an IPv6 access list.
<i>rule-num</i>	The number of a defined access list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

---

### Default

Packets matching this rule are forwarded.

---

### Usage Guidelines

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 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 access-list6 list-name rule rule-num description desc  
delete policy access-list6 list-name rule rule-num description  
show policy access-list6 list-name rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of an IPv6 access list.
<i>rule-num</i>	The number of a defined access list rule.
<i>desc</i>	A brief text description for the access list rule.

---

### Default

None.

---

### Usage Guidelines

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 access-list6 <list-name> rule <rule-num> destination

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

### Syntax

```
set policy access-list6 list-name rule rule-num destination {any | host ipv6 |  
inverse-mask ipv6 | network ipv6net}
```

```
delete policy access-list6 list-name rule rule-num destination
```

```
show policy access-list6 list-name rule rule-num destination
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {  
    access-list6 list-name {  
        rule rule-num {  
            destination {  
                any  
                host ipv6  
                inverse-mask ipv6  
                network ipv6net  
            }  
        }  
    }  
}
```

### Parameters

<i>list-name</i>	The name of an IPv6 access list.
<i>rule-num</i>	The number of a defined IPv6 access list.
<b>any</b>	Match packets destined for any destination. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>host <i>ipv6</i></b>	Match packets destined for the specified IPv6 host. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

<b>inverse-mask</b> <i>ipv6</i>	Match packets destined for the network specified by the mask. Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.
<b>network</b> <i>ipv6net</i>	Match packets destined for the specified network. The format is <i>ipv6-address/prefix</i> . Exactly one of <b>any</b> , <b>host</b> , <b>inverse-mask</b> , and <b>network</b> is mandatory.

---

---

### Default

None.

---

### Usage Guidelines

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

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

### Syntax

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

```
delete policy access-list6 list-name rule rule-num source
```

```
show policy access-list6 list-name rule rule-num source
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>list-name</i>	The name of an IPv6 access list.
<i>rule-num</i>	The number of a defined IPv6 access list rule.
<b>any</b>	Match packets coming from any source. Exactly one of <b>any</b> , <b>exact-match</b> , and <b>network</b> is mandatory.
<b>exact-match</b>	Match packets coming from one of the network prefixes. Exactly one of <b>any</b> , <b>exact-match</b> , and <b>network</b> is mandatory.
<b>network</b> <i>ipv6net</i>	Match packets coming from the specified network. The format is <i>ipv6p-address/prefix</i> . Exactly one of <b>any</b> , <b>exact-match</b> , and <b>network</b> is mandatory.

---

### Default

None.

---

### Usage Guidelines

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 as-path-list <list-name>

Defines an autonomous system (AS) path list.

---

### Syntax

```
set policy as-path-list list-name
delete policy as-path-list list-name
show policy as-path-list list-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	Multi-node. A text identifier for the AS path list. You can create multiple AS path lists by creating multiple <b>policy as-path-list</b> configuration nodes.
------------------	---

---

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy as-path-list list-name description desc  
delete policy as-path-list list-name description  
show policy as-path-list list-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined AS path list.
<i>desc</i>	A brief text description for the AS path list.

---

### Default

None.

---

### Usage Guidelines

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

Creates a rule for an AS path list.

---

### Syntax

```
set policy as-path-list list-name rule rule-num
delete policy as-path-list list-name rule rule-num
show policy as-path-list list-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	The name of a defined AS path list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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 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 as-path-list list-name rule rule-num action {deny | permit}
delete policy as-path-list list-name rule rule-num action
show policy as-path-list list-name rule rule-num action
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined AS path list.
<i>rule-num</i>	The number of a defined AS path list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

---

### Default

Packets matching this rule are forwarded.

---

### Usage Guidelines

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 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 as-path-list list-name rule rule-num description desc  
delete policy as-path-list list-name rule rule-num description  
show policy as-path-list list-name rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined AS path list.
<i>rule-num</i>	The number of a defined AS path list rule.
<i>desc</i>	A brief text description for the AS path list rule.

---

### Default

None.

---

### Usage Guidelines

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 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 as-path-list list-name rule rule-num regex regex
delete policy as-path-list list-name rule rule-num regex
show policy as-path-list list-name rule rule-num regex
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined AS path list.
<i>rule-num</i>	The number of a defined AS path list rule.
<i>regex</i>	A POSIX-style regular expression representing an AS path list.

---

### Default

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

---

### Usage Guidelines

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 as-path-list <list-name> rule <rule-num> action command`, 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 community-list <list-num>

Defines a BGP community list.

---

### Syntax

```
set policy community-list list-num
delete policy community-list list-num
show policy community-list list-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
    community-list list-num {}
}
```

---

### Parameters

<i>list-num</i>	Multi-node. A numeric identifier for the community list. You can create multiple community lists by creating multiple <b>policy community-list</b> configuration nodes.
-----------------	--

---

### Default

None.

---

### Usage Guidelines

Use the **set** form of this command to create a BGP community list for use in policy-based routing.

Use the **delete** form of this command to remove a community list.

Use the **show** form of this command to display community list configuration.

## policy community-list <list-num> description <desc>

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

---

### Syntax

```
set policy community-list list-num description desc
delete policy community-list list-num description
show policy community-list list-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  community-list list-num {
    description desc
  }
}
```

---

### Parameters

<i>list-num</i>	The number of a defined community list.
<i>desc</i>	A brief text description for the community list.

---

### Default

None.

---

### Usage Guidelines

Use the **set** form of this command to create a description for a community list.

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

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

## policy community-list <list-num> rule <rule-num>

Creates a rule for a community list.

---

### Syntax

```
set policy community-list list-num rule rule-num
delete policy community-list list-num rule rule-num
show policy community-list list-num rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  community-list list-num {
    rule rule-num {}
  }
}
```

---

### Parameters

---

<i>list-num</i>	The number of a defined community list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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

Use the **delete** form of this command to remove a community list rule.

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

## policy community-list <list-num> rule <rule-num> action

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

### Syntax

```
set policy community-list list-num rule rule-num action {deny | permit}
delete policy community-list list-num rule rule-num action
show policy community-list list-num rule rule-num action
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {
  community-list list-num {
    rule rule-num {
      action {
        deny
        permit
      }
    }
  }
}
```

### Parameters

<i>list-num</i>	The number of a defined community list.
<i>rule-num</i>	The number of a defined community list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

### Default

Packets matching this rule are forwarded.

---

### Usage Guidelines

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 community-list <list-num> rule <rule-num> description <desc>

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

---

### Syntax

```
set policy community-list list-num rule rule-num description desc  
delete policy community-list list-num rule rule-num description  
show policy community-list list-num rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {  
    community-list list-num {  
        rule rule-num {  
            description desc  
        }  
    }  
}
```

---

### Parameters

<i>list-num</i>	The number of a defined community list.
<i>rule-num</i>	The number of a defined community list rule.
<i>desc</i>	A brief text description for the community list rule.

---

### Default

None.

---

### Usage Guidelines

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

Use the **delete** form of this command to remove a community list rule description.

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

## policy community-list <list-num> rule <rule-num> regex <regex>

Defines match criteria for a community list rule based on a regular expression.

---

### Syntax

```
set policy community-list list-num rule rule-num regex regex
delete policy community-list list-num rule rule-num regex
show policy community-list list-num rule rule-num regex
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  community-list list-num {
    rule rule-num {
      regex regex
    }
  }
}
```

---

### Parameters

<i>list-num</i>	The number of a defined community list.
<i>rule-num</i>	The number of a defined community list rule.
<i>regex</i>	A POSIX-style regular expression representing a BGP community list.

---

### Default

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

---

### Usage Guidelines

Use the set form of this command to define the match criteria to be used to determine forwarding policy based on BGP community.

Packets are matched based on whether the communities listed in the packet match the regular expression defined using this command. Depending on the action defined for the rule using `policy community-list <list-num> rule <rule-num> action command`, 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 prefix-list <list-name>

Defines a prefix list.

---

### Syntax

```
set policy prefix-list list-name
delete policy prefix-list list-name
show policy prefix-list list-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	Multi-node. A text identifier for the prefix list. You can create multiple prefix lists by creating multiple <b>policy prefix-list</b> configuration nodes.
------------------	--

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy prefix-list list-name description desc
delete policy prefix-list list-name description
show policy prefix-list list-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>desc</i>	A brief text description for the prefix list.

---

### Default

None.

---

### Usage Guidelines

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

Creates a rule for a prefix list.

---

### Syntax

```
set policy prefix-list list-name rule rule-num
delete policy prefix-list list-name rule rule-num
show policy prefix-list list-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy prefix-list list-name rule rule-num action {deny | permit}
delete policy prefix-list list-name rule rule-num action
show policy prefix-list list-name rule rule-num action
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

---

### Default

Packets matching this rule are forwarded.



---

### Usage Guidelines

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

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

---

### Syntax

```
set policy prefix-list list-name rule rule-num description desc
delete policy prefix-list list-name rule rule-num description
show policy prefix-list list-name rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<i>desc</i>	A brief text description for the prefix list rule.

---

### Default

None.

---

### Usage Guidelines

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 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 prefix-list list-name rule rule-num ge value
```

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

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

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<i>value</i>	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.

---

### Default

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

---

### Usage Guidelines

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 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 prefix-list list-name rule rule-num le value
delete policy prefix-list list-name rule rule-num le
show policy prefix-list list-name rule rule-num le
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<i>value</i>	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.

---

### Default

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

---

### Usage Guidelines

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 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 prefix-list list-name rule rule-number prefix ipv4net
delete policy prefix-list list-name rule rule-num prefix
show policy prefix-list list-name rule rule-num prefix
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<i>ipv4net</i>	An IPv4 network. Networks exactly matching this network will match this rule. The format is <i>ip-address/prefix</i> .

---

### Default

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



---

### Usage Guidelines

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 prefix-list6 <list-name>

Defines an IPv6 prefix list.

---

### Syntax

```
set policy prefix-list6 list-name
delete policy prefix-list6 list-name
show policy prefix-list6 list-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	Multi-node. A text identifier for the IPv6 prefix list. You can create multiple IPv6 prefix lists by creating multiple <b>policy prefix-list6</b> configuration nodes.
------------------	---

---

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy prefix-list6 list-name description desc
delete policy prefix-list6 list-name description
show policy prefix-list6 list-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined IPv6 prefix list.
<i>desc</i>	A brief text description for the prefix list.

---

### Default

None.

---

### Usage Guidelines

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

Creates a rule for an IPv6 prefix list.

---

### Syntax

```
set policy prefix-list6 list-name rule rule-num
delete policy prefix-list6 list-name rule rule-num
show policy prefix-list6 list-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>list-name</i>	The name of a defined IPv6 prefix list.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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 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 prefix-list6 list-name rule rule-num action {deny | permit}
delete policy prefix-list6 list-name rule rule-num action
show policy prefix-list6 list-name rule rule-num action
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>list-name</i>	The name of a defined IPv6 prefix list.
<i>rule-num</i>	The number of a defined IPv6 prefix list rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

### Default

Packets matching this rule are forwarded.

---

### Usage Guidelines

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 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 prefix-list6 list-name rule rule-num description desc  
delete policy prefix-list6 list-name rule rule-num description  
show policy prefix-list6 list-name rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

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

---

### Default

None.

---

### Usage Guidelines

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 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 prefix-list6 list-name rule rule-num ge value
```

```
delete policy prefix-list6 list-name rule rule-num ge
```

```
show policy prefix-list6 list-name rule rule-num ge
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined IPv6 prefix list.
<i>rule-num</i>	The number of a defined IPv6 prefix list rule.
<i>value</i>	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.

---

### Default

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

---

### Usage Guidelines

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 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 prefix-list6 list-name rule rule-num le value
delete policy prefix-list6 list-name rule rule-num le
show policy prefix-list6 list-name rule rule-num le
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined IPv6 prefix list.
<i>rule-num</i>	The number of a defined IPv6 prefix list rule.
<i>value</i>	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.

---

### Default

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

---

### Usage Guidelines

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 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 prefix-list6 list-name rule rule-number prefix ipv6net
delete policy prefix-list6 list-name rule rule-num prefix
show policy prefix-list6 list-name rule rule-num prefix
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>list-name</i>	The name of a defined prefix list.
<i>rule-num</i>	The number of a defined prefix list rule.
<i>ipv6net</i>	An IPv6 network. Networks exactly matching this network will match this rule. The format is <i>ipv6-address/prefix</i> (that is < <i>x::x::x::x::x::x</i> >/<0-128>).

---

### Default

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

---

### Usage Guidelines

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-map <map-name>

Defines a route map for policy-based routing.

---

### Syntax

```
set policy route-map map-name
delete policy route-map map-name
show policy route-map map-name
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>map-name</i>	Multi-node. A text identifier for the route map. You can create multiple route maps by creating multiple <b>policy route-map</b> configuration nodes.
-----------------	--

---

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> description <desc>

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

---

### Syntax

```
set policy route-map map-name description desc
delete policy route-map map-name description
show policy route-map map-name description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>desc</i>	A brief text description for the route map.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num>

Creates a rule for a route map.

---

### Syntax

```
set policy route-map map-name rule rule-num
delete policy route-map map-name rule rule-num
show policy route-map map-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	Multi-node. A numeric identifier for the rule. The range is 1 to 4294967295.  You can define multiple rules by creating multiple <b>rule</b> configuration nodes.

---

---

### Default

None.

---

### Usage Guidelines

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.

## policy 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-map map-name rule rule-num action {deny | permit}
delete policy route-map map-name rule rule-num action
show policy route-map map-name rule rule-num action
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>deny</b>	Packets matching this rule are silently dropped.
<b>permit</b>	Packets matching this rule are forwarded.

### Default

Routes are denied.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> call <target>

Calls to another route map.

---

### Syntax

```
set policy route-map map-name rule rule-num call target
delete policy route-map map-name rule rule-num call
show policy route-map map-name rule rule-num
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>target</i>	The identifier of the route map being called.

---

### Default

None.

---

### Usage Guidelines

Use the set form of this command to call to another route map.

The new route map is called after all **set** actions specified in the route map have been performed. If the called route map returns **permit**, then the matching and exit policies of the route map govern further behavior in the normal way. If the called route-map returns **deny**, processing of the route map completes and the route is denied, regardless of any further matching or exit policies.

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

Calls to another rule within the current route map.

---

### Syntax

```
set policy route-map map-name rule rule-num continue target-num
delete policy route-map map-name rule rule-num continue
show policy route-map map-name rule rule-num continue
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>target</i>	The identifier of the route map rule being called.

---

### Default

None.

---

### Usage Guidelines

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

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

---

### Syntax

```
set policy route-map map-name rule rule-num description desc  
delete policy route-map map-name rule rule-num description  
show policy route-map map-name rule rule-num description
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

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

---

### Default

None.

---

### Usage Guidelines

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-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-map map-name rule rule-num match as-path list-name
delete policy route-map map-name rule rule-num match as-path
show policy route-map map-name rule rule-num match as-path
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>list-name</i>	Matches the AS paths in the route with those permitted by the specified AS path list. The AS path list must already be defined.

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-map <map-name> rule <rule-num> match community

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

### Syntax

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

```
delete policy route-map map-name rule rule-num match community
```

```
show policy route-map map-name rule rule-num match community
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      match {
        community {
          community-list list-num
          exact-match
        }
      }
    }
  }
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>community-list</b> <i>list-num</i>	Matches the BGP communities in the route with those permitted by the specified community list. The community list policy must already be defined. Either <b>community-list</b> or <b>exact-match</b> must be specified.

---

<b>exact-match</b>	BGP communities are to be matched exactly. Either <b>community-list</b> or <b>exact-match</b> must be specified.
--------------------	--

---

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-map <map-name> rule <rule-num> match interface <ethx>

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

---

### Syntax

```
set policy route-map map-name rule rule-num match interface ethx
delete policy route-map map-name rule rule-num match interface
show policy route-map map-name rule rule-num match interface
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>ethx</i>	Matches first hop interface specified in the route against the interface name.

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-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-map map-name rule rule-num match ip address {access-list list-num
| prefix-list list-name}
```

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

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

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>access-list</b> <i>list-num</i>	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 <b>access-list</b> or <b>prefix-list</b> must be specified.



---

<b>prefix-list</b> <i>list-name</i>	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 <b>access-list</b> or <b>prefix-list</b> must be specified.
--	--

---

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-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-map map-name rule rule-num match ip nexthop {access-list list-num
| prefix-list list-name}
```

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

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

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
access-list <i>list-num</i>	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 <b>access-list</b> or <b>prefix-list</b> must be specified.

---

<b>prefix-list</b> <i>list-name</i>	Matches next-hop IP address in the route against those permitted by the specified prefix list. The prefix list must already be defined. Either <b>access-list</b> or <b>prefix-list</b> must be specified.
-------------------------------------	--

---

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-map <map-name> rule <rule-num> match ip route-source

Defines a match condition for a route map based on the address from where a route is advertised.

### Syntax

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

```
delete policy route-map map-name rule rule-num match ip route-source
```

```
show policy route-map map-name rule rule-num match ip route-source
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
access-list <i>list-num</i>	Matches routes advertised from addresses contained in the specified access list. The access list must already be defined. Either <b>access-list</b> or <b>prefix-list</b> must be specified.

---

<b>prefix-list</b> <i>list-name</i>	Matches routes advertised from addresses contained in the specified prefix list. The prefix list must already be defined. Either <b>access-list</b> or <b>prefix-list</b> must be specified.
-------------------------------------	--

---

---

### Default

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

---

### Usage Guidelines

Use the **set** form of this command to define a match condition for a route map policy based on the address from where routes are advertised (its route source).

Packets are matched based on whether the route source matches an address contained in the specified access list or prefix list. Depending on the action defined for the rule using `policy route-map <map-name> rule <rule-num> action command`, 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-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-map map-name rule rule-num match ipv6 address {access-list6
list-num | prefix-list6 list-name}
```

```
delete policy route-map map-name rule rule-num match ipv6 address
```

```
show policy route-map map-name rule rule-num match ipv6 address
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      match {
        ipv6 address {
          access-list6 list-num
          prefix-list6 list-name
        }
      }
    }
  }
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>access-list6</b> <i>list-num</i>	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 <b>access-list6</b> or <b>prefix-list6</b> must be specified.

---

<b>prefix-list6</b> <i>list-name</i>	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 <b>access-list6</b> or <b>prefix-list6</b> must be specified.
---	--

---

---

### Default

If no IPv6 address match condition is specified, packets are not filtered by IPv6 address.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command**, 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-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-map map-name rule rule-num match ipv6 nexthop {access-list6  
list-num | prefix-list6 list-name}
```

```
delete policy route-map map-name rule rule-num match ipv6 nexthop
```

```
show policy route-map map-name rule rule-num match ipv6 nexthop
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {  
    route-map map-name {  
        rule rule-num {  
            match {  
                ipv6 {  
                    nexthop {  
                        access-list6 list-num  
                        prefix-list6 list-name  
                    }  
                }  
            }  
        }  
    }  
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>access-list6</b> <i>list-num</i>	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 <b>access-list6</b> or <b>prefix-list6</b> must be specified.



---

<b>prefix-list6</b> <i>list-name</i>	Matches next-hop IPv6 address in the route against those permitted by the specified prefix list. The prefix list must already be defined. Either <b>access-list6</b> or <b>prefix-list6</b> must be specified.
---	--

---

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-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-map map-name rule rule-num match metric metric
delete policy route-map map-name rule rule-num match metric
show policy route-map map-name rule rule-num match metric
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>metric</i>	A number representing a route metric. This value is matched against the metric in the route.

---

### Default

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

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-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-map map-name rule rule-num match origin {egp | igp | incomplete}
delete policy route-map map-name rule rule-num match origin
show policy route-map map-name rule rule-num match origin
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      match {
        origin {
          origin-code [egp|igp|incomplete]
        }
      }
    }
  }
}
```

---

### Parameters

---

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>egp:</b>	Matches routes whose origin is an Exterior Gateway Protocol.
<b>igp:</b>	Matches routes whose origin is an Interior Gateway Protocol.
<b>incomplete</b>	Matches routes whose BGP origin code is incomplete.

---

---

### Default

If no origin match condition is specified, packets are not filtered by BGP origin code.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> action command`, 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-map <map-name> rule <rule-num> match peer <ipv4>

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

---

### Syntax

```
set policy route-map map-name rule rule-num match peer ipv4
delete policy route-map map-name rule rule-num match peer
show policy route-map map-name rule rule-num match peer
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      match {
        peer ipv4
      }
    }
  }
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>ipv4</i>	An IPv4 address. This address is matched against the peer address in the route.

---

### Default

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

---

### Usage Guidelines

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

Packets are matched based on whether the address of the peer in the route matches that specified by this command. Depending on the action defined for the rule using `policy route-map <map-name> rule <rule-num> action command`, 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 peer address match condition.

Use the **show** form of this command to display peer address match condition configuration.

## policy 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-map map-name rule rule-num match tag tag
delete policy route-map map-name rule rule-num match tag
show policy route-map map-name rule rule-num match tag
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>tag</i>	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.

---

### Default

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



---

### Usage Guidelines

Use the **set** form of this command to define a match condition for a route map policy based on OSPF tag.

Packets are matched based on whether the value of the OSPF external LSA 32-bit tag field in the route matches that specified by this command. Depending on the action defined for the rule using `policy route-map <map-name> rule <rule-num> action command`, 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 OSPF tag match condition.

Use the **show** form of this command to display OSPF tag match condition configuration.

## policy route-map <map-name> rule <rule-num> on-match

Specifies an alternative exit policy for a route map.

---

### Syntax

```
set policy route-map map-name rule rule-num on-match {goto rule-num | next}  
delete policy route-map map-name rule rule-num on-match  
show policy route-map map-name rule rule-num on-match
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {  
  route-map map-name {  
    rule rule-num {  
      on-match {  
        goto rule-num  
      }  
    }  
  }  
}
```

---

### Parameters

---

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>goto</b> <i>rule-num</i>	The number of a defined route map rule. When all matches listed in the route map rule succeed, the current route map rule is exited and this rule is invoked and executed. Note that jumping to a previous route map rule is not permitted.
<b>next</b>	When all matches listed in the route map rule succeed, the current route map rule is exited and the next rule in the sequence is invoked and executed.

---

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set aggregator

Modifies the BGP aggregator attribute of a route.

### Syntax

```
set policy route-map map-name rule rule-num set aggregator {as asn | ip ipv4}  
delete policy route-map map-name rule rule-num set aggregator  
show policy route-map map-name rule rule-num set
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {  
  route-map map-name {  
    rule rule-num {  
      set {  
        aggregator {  
          as asn  
          ip ipv4  
        }  
      }  
    }  
  }  
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
as <i>asn</i>	Modifies the autonomous system number of the BGP aggregator in the route to the specified value. The range is 1 to 65535.
ip <i>ipv4</i>	Modifies the IP address of the BGP aggregator in the route to the specified IPv4 address.

---

### Default

None.

---

### Usage Guidelines

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-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-map map-name rule rule-num set as-path-prepend prepend
delete policy route-map map-name rule rule-num set as-path-prepend
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      set {
        as-path-prepend prepend
      }
    }
  }
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>prepend</i>	A string representing an AS path.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set atomic-aggregate

Sets the BGP atomic-aggregate attribute in a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set atomic-aggregate
delete policy route-map map-name rule rule-num set atomic-aggregate
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      set {
        atomic-aggregate
      }
    }
  }
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set comm-list

Modifies the BGP community list in a route.

### Syntax

```
set policy route-map map-name rule rule-num set comm-list {comm-list list-name | delete}
```

```
delete policy route-map map-name rule rule-num set comm-list
```

```
show policy route-map map-name rule rule-num set
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      set {
        comm-list {
          comm-list list-name
          delete
        }
      }
    }
  }
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>comm-list</b> <i>list-name</i>	Removes the communities in the specified community list from the route's community list. The community list must already be defined.
<b>delete</b>	Deletes the route's entire community list.

---

**Default**

None.

---

**Usage Guidelines**

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.

## policy route-map <map-name> rule <rule-num> set community

Modifies the BGP communities attribute in a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set community { “[additive]
community” | none }
```

```
delete policy route-map map-name rule rule-num set community
```

```
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

---

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>additive</b>	Appends the specified community to the existing communities in the route. Double-quotes must be used when <b>additive</b> is specified.
<i>community</i>	A BGP community. Supported values are a community number in <i>aa:nn</i> format, or the well-known BGP communities <b>local-AS</b> , <b>no-export</b> , <b>no-advertise</b> , or <b>internet</b> .
<b>none</b>	Remove communities attribute from BGP updates.

---

---

### Default

When the **additive** keyword is not used, the specified community replaces the existing communities in the route.

---

### Usage Guidelines

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.

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-map <map-name> rule <rule-num> set ip-next-hop <ipv4>

Modifies the next hop destination of a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set ip-next-hop ipv4
delete policy route-map map-name rule rule-num set ip-next-hop
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      set {
        ip-next-hop ipv4
      }
    }
  }
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>ip-next-hop ipv4</i>	The IPv4 address of the next hop.

---

### Default

None.

---

### Usage Guidelines

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-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-map map-name rule rule-num set ipv6-next-hop {global | local} ipv6  
delete policy route-map map-name rule rule-num set ipv6-next-hop {global | local}  
show policy route-map map-name rule rule-num set
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {  
  route-map map-name {  
    rule rule-num {  
      set {  
        ipv6-next-hop {  
          global ipv6  
          local ipv6  
        }  
      }  
    }  
  }  
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<b>global</b>	The next hop address is an IPv6 global address.
<b>local</b>	The next hop address is an IPv6 local address.
<i>ipv6</i>	The IPv6 address of the next hop.



---

### Default

None.

---

### Usage Guidelines

Use this command to modify the IPv6 next hop destination address 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 **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-map <map-name> rule <rule-num> set local-preference <local-pref>

Modifies the BGP local-pref attribute in a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set local-preference local-pref
delete policy route-map map-name rule rule-num set local-preference
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {
  route-map map-name {
    rule rule-num {
      set {
        local-preference local-pref
      }
    }
  }
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>local-pref</i>	The new value for the BGP local preference path attribute.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set metric <metric>

Modifies the metric of a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set metric metric  
delete policy route-map map-name rule rule-num set metric  
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>metric</i>	A number representing the new metric to be used in the route.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set metric-type <type>

Specifies the OSPF external metric-type for a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set metric-type type  
delete policy route-map map-name rule rule-num set metric-type  
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {  
  route-map map-name {  
    rule rule-num {  
      set {  
        metric-type type  
      }  
    }  
  }  
}
```

---

### Parameters

---

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>type-1</i>	OSPF external type 1 metric. This metric uses both internal and external costs when calculating the cost to access an external network.
<i>type-2</i>	OSPF external type 2 metric. This metric uses only external cost when calculating the cost to access an external network.

---

---

### Default

None.

---

### Usage Guidelines

Use this command to specify the metric OSPF should use to calculate 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-map <map-name> rule <rule-num> set origin

Modifies the BGP origin code of a route.

### Syntax

```
set policy route-map map-name rule rule-num set origin {asn | egp | igp | incomplete}  
delete policy route-map map-name rule rule-num set origin  
show policy route-map map-name rule rule-num set
```

### Command Mode

Configuration mode.

### Configuration Statement

```
policy {  
    route-map map-name {  
        rule rule-num {  
            set {  
                origin [egp|igp|incomplete]  
            }  
        }  
    }  
}
```

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>asn</i>	An autonomous system number. The range is 1 to 65535.
<b>egp</b>	Sets the BGP origin code to <b>egp</b> (Exterior Gateway Protocol).
<b>igp</b>	Sets the BGP origin code to <b>igp</b> (Interior Gateway Protocol).
<b>incomplete</b>	Sets the BGP origin code to <b>incomplete</b> .



---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set originator-id <ipv4>

Modifies the BGP originator ID attribute of a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set originator-id ipv4  
delete policy route-map map-name rule rule-num set originator-id  
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

```
policy {  
  route-map map-name {  
    rule rule-num {  
      set {  
        originator-id ipv4  
      }  
    }  
  }  
}
```

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>ipv4</i>	The IPv4 address to be used as the new originator ID.

---

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set tag <tag>

Modifies the OSPF tag value of a route.

### Syntax

```
set policy route-map map-name rule rule-num set tag tag  
delete policy route-map map-name rule rule-num set tag  
show policy route-map map-name rule rule-num set
```

### Command Mode

Configuration mode.

### Configuration Statement

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

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>tag</i>	A 32-bit number representing the new value of the OSPF external Link-State Advertisement (LSA) tag field.

### Default

None.

---

### Usage Guidelines

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-map <map-name> rule <rule-num> set weight <weight>

Modifies the BGP weight of a route.

---

### Syntax

```
set policy route-map map-name rule rule-num set weight weight
delete policy route-map map-name rule rule-num set weight
show policy route-map map-name rule rule-num set
```

---

### Command Mode

Configuration mode.

---

### Configuration Statement

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

---

### Parameters

<i>map-name</i>	The name of a defined route map.
<i>rule-num</i>	The number of a defined route map rule.
<i>weight</i>	The BGP weight to be recorded in the routing table. The range is 0 to 65535.

---

### Default

None.

---

### Usage Guidelines

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
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to display IP access lists.

---

### Examples

[Example 3-1](#) shows IP access lists.

Example 3-1 “show ip access-list”: Displaying 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
vyatta@vyatta:~$
```





---

## show ip as-path-access-list

Displays all as-path access lists.

---

### Syntax

```
show ip as-path-access-list
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to display as-path access lists.

---

### Examples

[Example 3-2](#) shows as-path access lists.

Example 3-2 “show ip as-path-access-list”: Displaying 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
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to display community lists.

---

### Examples

[Example 3-3](#) shows community lists.

Example 3-3 “show ip community-list”: Displaying 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
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to display extended IP community lists.

---

### Examples

[Example 3-4](#) shows extended IP community lists.

Example 3-4 “show ip extcommunity-list”: Displaying 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]]]
```

---

### Command Mode

Operational mode.

---

### Parameters

<b>detail</b>	Displays detailed information for all IP prefix lists.
<b>summary</b>	Displays summary information for all IP prefix lists.
<i>list-name</i>	Displays information about the named IP prefix list.
<i>seq-num</i>	Displays the specified sequence from the named IP prefix list.
<i>ipv4net</i>	Displays the select prefix of the named IP prefix list.
<b>first-match</b>	Displays the first match from the select prefix of the named IP prefix list.
<b>longer</b>	Displays the longer match of the select prefix from the named IP prefix list.

---

### Default

None.

---

### Usage Guidelines

Use this command to display prefix lists.

---

### Examples

[Example 3-5](#) shows prefix lists.

---

**Example 3-5 “show ip prefix-list”: Displaying 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
```

---

## Command Mode

Operational mode.

---

## Parameters

None.

---

## Default

None.

---

## Usage Guidelines

Use this command to display IP route maps per protocol.

---

## Examples

[Example 3-6](#) shows IP route maps by protocol.

Example 3-6 “show ip protocol”: Displaying 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]
```

---

### Command Mode

Operational mode.

---

### Parameters

None.

---

### Default

None.

---

### Usage Guidelines

Use this command to display route map information.

---

### Examples

[Example 3-7](#) shows route map information.

Example 3-7 “show route-map”: Displaying route map information

---

```
vyatta@vyatta:~$ show route-map
ZEBRA:
route-map MAP1, permit, sequence 1
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
RIP:
route-map MAP1, permit, sequence 1
  Match clauses:
    interface eth0
  Set clauses:
  Call clause:
```

```
    Action:
      Exit routemap
OSPF:
route-map MAP1, permit, sequence 1
  Match clauses:
    interface eth0
  Set clauses:
  Call clause:
  Action:
    Exit routemap
BGP:
route-map MAP1, permit, sequence 1
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
vyatta@vyatta:~$
```

---

## Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
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
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
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
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
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
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

---

---

PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
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
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
SSH	Secure Shell
SSID	Service Set Identifier
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

---