Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

**Text formatting conventions**

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bold text</td>
<td>Identifies command names</td>
</tr>
<tr>
<td></td>
<td>Identifies keywords and operands</td>
</tr>
<tr>
<td></td>
<td>Identifies the names of user-manipulated GUI elements</td>
</tr>
<tr>
<td></td>
<td>Identifies text to enter at the GUI</td>
</tr>
<tr>
<td>italic text</td>
<td>Identifies emphasis</td>
</tr>
<tr>
<td></td>
<td>Identifies variables</td>
</tr>
<tr>
<td></td>
<td>Identifies document titles</td>
</tr>
<tr>
<td>Courier font</td>
<td>Identifies CLI output</td>
</tr>
<tr>
<td></td>
<td>Identifies command syntax examples</td>
</tr>
</tbody>
</table>

**Command syntax conventions**

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bold text</td>
<td>Identifies command names, keywords, and command options.</td>
</tr>
<tr>
<td>italic text</td>
<td>Identifies a variable.</td>
</tr>
<tr>
<td>value</td>
<td>In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, --show WWN.</td>
</tr>
</tbody>
</table>
Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

**NOTE**
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

**ATTENTION**
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

**CAUTION**
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

**DANGER**
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.
Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads. White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

<table>
<thead>
<tr>
<th>Online</th>
<th>Telephone</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred method of contact for non-urgent issues:</td>
<td>Required for Sev 1-Critical and Sev 2-High issues:</td>
<td><a href="mailto:support@brocade.com">support@brocade.com</a></td>
</tr>
<tr>
<td>- My Cases through MyBrocade</td>
<td>- Continental US: 1-800-752-8061</td>
<td>Please include:</td>
</tr>
<tr>
<td>- Software downloads and licensing tools</td>
<td>- Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33)</td>
<td>- Problem summary</td>
</tr>
<tr>
<td>- Knowledge Base</td>
<td>- For areas unable to access toll free number: +1-408-333-6061</td>
<td>- Serial number</td>
</tr>
<tr>
<td></td>
<td>- Toll-free numbers are available in many countries.</td>
<td>- Installation details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Environment description</td>
</tr>
</tbody>
</table>

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
'Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.

• For questions regarding service levels and response times, contact your OEM/Solution Provider.
Policy-Based Routing

• Introduction ....................................................................................................................... 9

Introduction

Policy-based routing (PBR) enables you to use IP traffic rules to classify traffic based on its attributes and apply processing differentially according to the classification, and to selectively route IP packets, for example, to an alternate next hop. PBR on the Vyatta system is supported just on incoming Layer 3 and Layer 4 traffic.

All packets received on an interface are considered for policy-based routing provided that interface is assigned a routing policy.

When no routing policies are applied, routing decisions are made by using the default (main) routing table (Table 254) of the system.

PBR policies can be applied to dataplane interfaces for inbound traffic, but not to loopback, tunnel, bridge, OpenVPN, VTI, and IP unnumbered interfaces.

On the Vyatta system, you cannot apply policy based routing to locally generated packets.

Defining a routing policy

The routing policy classifies traffic and specifies the handling that should take place for different classes. This classification and handling are accomplished by using a set of policy rules.

Rules are configured with match criteria that include an extensive set of attributes—including protocol, source and destination addresses and ports, fragmentation, ICMP or ICMPv6 type, and TCP flags. You can also preconfigure groups of addresses, ports, and networks and refer to these groups in policy rules.

The routing policy must be applied to an interface for the policy to be effective.

To implement policy-based routing, perform the following steps:

1. Define the policy rules.
2. Attach the policy to an ingress interface.
3. Create a route in a PBR table other than Table 254.

NOTE
Table 254 is also known as the main table or default table.

Routing policy rules

Packets that match the PBR rule criteria do one of the following:

• They are dropped (if the drop action is set).
• They are routed by using a specific PBR routing table.
Packets that match the rule parameters are considered for policy-based routing. As many as 9,999 rules in a policy are supported. If no match criteria are specified, all packets are routed according to the default Table 254.

The packets that do not match any policy rule are routed according to the routes in the main table.

Routing policy rules are executed in numeric sequence, from lowest to highest. You can renumber rules by using the `rename` command in configuration mode (refer to Vyatta Basic System Reference Guide).

---

**NOTE**

To avoid having to renumber routing policy rules, a good practice is to number rules in increments of 10. This increment allows room for the insertion of new rules within the policy.

---

**PBR behavior**

Routes that remain persistent in the controller. If the dataplane goes down, and up, the routes are automatically re-established without the need for reconfiguration.

PBR does not reassemble fragmented packets. PBR treats fragments as individual packets.

PBR rules can be changed dynamically and does not require the rebinding of the PBR policy to an interface.

Configuration for VLAN-based classification, virtual interface (vif), MAC address, packet mangling, and so on, are not supported.

The controller automatically continuously resyncs the route information to the dataplane.

Multiple PBR policies can be applied to an interface. For best results, we recommend that these policies are unique.

---

**Packet forwarding path**

When enabled, PBR processes incoming packets after packet validation and firewall action. Packets received by the dataplane ingress interfaces for transmission to the egress interface follow the forwarding path listed below. There is only a single Virtual Routing and Forwarding (VRF) instance for PBR.

1. Packet validation and reassembly
2. Firewall
3. PBR classification, route table ID determination
4. NAT
5. Firewall
6. QoS
7. Transmit out of an egress interface
PBR routing example

The following figure shows a simple site that uses PBR on the Vyatta system (R1) to route traffic from two different internal subnets to two Internet links.

The following conditions apply to this scenario:

- All Internet-bound traffic from subnet 192.168.10.0/24 is routed out interface dp0p1p1.
- All Internet-bound traffic from subnet 192.168.20.0/24 is routed out interface dp0p1p2.

To configure the scenario, perform the following steps in configuration mode.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
</table>
| Create Rule 10 and specify the destination address to match. In this case, any destination address is a match. | vyatta@R1# set policy route pbr myroute rule 10 action accept  
vyatta@R1# set policy route pbr myroute rule 10 destination address 0.0.0.0/0 |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the source address to match. In this case, any address on subnet 192.168.10.0/24 is a match.</td>
<td>vyatta@R1# set policy route pbr myroute rule 10 source address 192.168.10.0/24</td>
</tr>
<tr>
<td>Specify that all matching packets use alternate routing table 1.</td>
<td>vyatta@R1# set policy route pbr myroute rule 10 table 1</td>
</tr>
</tbody>
</table>
| Create rule 20 and specify the destination address to match. In this case, any destination address is a match.  | vyatta@R1# set policy route pbr myroute rule 20 address-family ipv4  
vyatta@R1# set policy route pbr myroute rule 20 action accept  
vyatta@R1# set policy route pbr myroute rule 20 destination address 0.0.0.0/0 |
| Specify the source address to match. In this case, any address on subnet 192.168.20.0/24 is a match. | vyatta@R1# set policy route pbr myroute rule 20 source address 192.168.20.0/24 |
| Specify that all matching packets use alternate routing table 2.      | vyatta@R1# set policy route pbr myroute rule 20 table 2                   |
| Commit the changes.                                                   | vyatta@R1# commit                                                        |
| Show the policy-based routing configuration.                         | vyatta@R1# show policy route                                                 |
|                                                                      | route {                                                                     |
|                                                                      |   pbr myroute {                                                             |
|                                                                      |     rule 10 {                                                              |
|                                                                      |       action accept                                                        |
|                                                                      |       destination {                                                        |
|                                                                      |         address 0.0.0.0/0                                                  |
|                                                                      |       }                                                                    |
|                                                                      |     source {                                                              |
|                                                                      |       address 192.168.10.0/24                                             |
|                                                                      |     }                                                                    |
|                                                                      |   }                                                                      |
|                                                                      | }                                                                         |
| Create the alternative routing table 1.                              | vyatta@R1# set protocols static table 1 route 12.34.56.0/24 next-hop 12.34.56.11 |
| Create the alternative routing table 2.                              | vyatta@R1# set protocols static table 2 route 12.34.56.0/24 next-hop 98.76.54.22 |
| Commit the change.                                                   | vyatta@R1# commit                                                        |
### TABLE 1  Routing using PBR (Continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
</table>
| Show the alternate routing table configuration. | vyatta@R1# show protocols static

```plaintext
static {
  table 1 {
    route 1.2.3.0/24 {
      next-hop 12.34.56.11 {
      }
    }
  }
  table 2 {
    route 2.3.4.0/24 {
      next-hop 98.76.54.22 {
      }
    }
  }
}
```

| Apply the 12.34.56.33/24 address to dp0p1p1. | vyatta@R1# set interfaces dataplane dp0p1p1 address 12.34.56.33/24 |

| Show the dataplane interface configuration. | vyatta@R1# show interfaces dataplane

```plaintext
dataplane dp0p1p1 {
  address 12.34.56.33/24
}
dataplane dp0p1p2 {
  address 98.76.54.44/24
}
dataplane dp0p1p3 {
  address 192.168.10.254/24
  policy {
    route myroute
  }
}
dataplane dp0p1p4 {
  address 192.168.20.254/24
  policy {
    route myroute
  }
}
```

### Binding interfaces to PBR tables

To configure an interface-based static route in a policy route table, perform the following steps:

#### TABLE 2  Applying a policy route to an interface

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the interface route for the interface.</td>
<td>vyatta@R1# set protocols static table 10 interface-route 192.168.20.254/24 next-hop-interface dp0p256p1 distance 25</td>
</tr>
</tbody>
</table>

| View the configuration. | vyatta@vyatta:-$ show protocols

```plaintext
protocols {
  static {
    table 10 {
      interface-route 192.168.20.254/24 {
        next-hop-interface dp0p256p1 {
        distance 1
      }
    }
  }
}
```

---

**Policy-Based Routing Reference Guide**  
53-1003723-02

---
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply the policy route to the interface.</td>
<td>vyatta@R1# set interfaces dataplane dp0p16p1 policy route myroute</td>
</tr>
<tr>
<td>Commit the change.</td>
<td>vyatta@R1# commit</td>
</tr>
</tbody>
</table>
Policy-Based Routing Commands

- clear policy......................................................................................................................16
- interfaces <interface> policy route <name>........................................................................17
- policy route pbr name <name> rule <number>...................................................................18
- policy route pbr name <name> rule <number> action <action>........................................19
- policy route pbr name <name> rule <number> af <protocol>............................................20
- policy route pbr name <name> rule <number> description <description>.......................21
- policy route pbr name <name> rule <number> destination <destination>.........................22
- policy route pbr name <name> rule <number> disable......................................................24
- policy route pbr name <name> rule <number> icmp <icmp>............................................25
- policy route pbr name <name> rule <number> icmpv6 <icmpv6>....................................26
- policy route pbr name <name> rule <number> log..........................................................27
- policy route pbr name <name> rule <number> port <port-name>.....................................28
- policy route pbr name <name> rule <number> protocol <protocol>.................................30
- policy route pbr name <name> rule <number> set <match-criteria>.................................32
- policy route pbr name <name> rule <number> source address <address>......................33
- policy route pbr name <name> rule <number> source port <port>....................................35
- policy route pbr name <name> rule <number> table..........................................................37
- policy route pbr name <name> rule <number> tcp flags <flags>....................................38
- show policy route <interface>......................................................................................39
- Related commands.......................................................................................................39
clear policy

Clears the statistics for route policies.

Syntax  clear policy

Modes   Operational mode

Usage Guidelines  Use this command to clear the statistics for policy-based routing.
```
interfaces <interface> policy route <name>

Applies an IP routing policy to inbound traffic on an interface.

**Syntax**
```
set interfaces interface policy route name
```
```
delete interfaces interface policy route [ name ]
```
```
show interfaces interface policy route
```

**Parameters**
```
interface
```
The type of interface. For detailed keywords and arguments that can be specified as interface types, refer to **Supported Interface Types** on page 45

---

**NOTE**
Policy-based routing policies can be applied to dataplane interfaces, but not on loopback, tunnel, bridge, OpenVPN, or VTI interfaces.

```
nome
```
An IP routing policy.

**Modes**
Configuration mode

**Configuration Statement**
```
interfaces interface {
    policy {
        route name
    }
}
```

**Usage Guidelines**
A routing policy has no effect on traffic traversing the system until it has been applied to an interface. To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr name name rule number` command, then apply the routing policy to interfaces by using a statement like this one. Once applied, the rule set acts as a packet filter. Use the `set` form of this command to apply an IP routing policy to an interface. Use the `delete` form of this command to remove an IP routing policy from an interface. Use the `show` form of this command to display an IP routing policy configuration for an interface.
policy route pbr name <name> rule <number>

Defines an IP routing policy rule.

Syntax

set policy route pbr name name rule number

delete policy route pbr name name rule [ number ]

show policy route pbr name name rule

Parameters

name

The name of an IP routing policy.

number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one rule configuration node.

Modes

Configuration mode

Configuration Statement

policy {
  route {
    pbr {
      name name {
        rule number
      }
    }
  }
}

Usage Guidelines

A policy identifies traffic that matches parameters and specifies which routing table to use. The table defines the route for a packet to take. A routing policy is a named collection of as many as 9,999 packet-classification rules. When applied to an interface, the policy rule classifies incoming traffic.

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

Use the delete form of this command to delete an existing IP routing policy.

Use the show form of this command to display a rule.
**policy route pbr name <name> rule <number> action <action>**

Defines the action for an IP routing policy rule.

**Syntax**

```
set policy route pbr name name rule number action { drop | accept }
delete policy route pbr name name rule number action [ drop | accept ]
show policy route pbr name name rule number action
```

**Parameters**

- `name`  The name of an IP routing policy.
- `number`  The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.
  - You can define multiple rules by creating more than one rule configuration node.
- `accept`  Accepts the packet.
- `drop`  Drops the packet silently.

**Modes**  Configuration mode

**Configuration Statement**

```
policy {
  route {
    pbr name name {
      rule number {
        action accept drop
      }
    }
  }
}
```

**Usage Guidelines**

If a rule does not explicitly drop a packet in the action, the PBR action is to accept the packet, which causes it to be sent to the specified alternate routing table for lookup and forwarding.

An applied policy can only be deleted after first removing it from an assigned interface.

Use the `set` form of this command to set the action for a rule.

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

Use the `show` form of this command to display a rule within an IP routing policy.
policy route pbr name <name> rule <number> af <protocol>

Defines the address family and routing protocol for an IP routing policy rule.

**Syntax**

```plaintext
set policy route pbr name name rule rule-number af [ipv4 | ipv6]
delete policy route pbr name name rule rule-number af [ipv4 | ipv6]
show policy route pbr name name rule rule-number af
```

**Parameters**

- `name`
  - The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

- `number`
  - The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999. You can define multiple rules by creating more than one rule configuration node.

**Modes**

Configuration mode

```plaintext
policy {
  route {
    pbr {
      name name {
        rule number {
          address-family
            ipv4
            ipv6
        }
      }
    }
  }
}
```

**Usage Guidelines**

Use the **set** form of this command to define the address family and routing protocol for an IP routing policy rule.

Use the **delete** form of this command to remove the address family and routing protocol for an IP routing policy rule.

Use the **show** form of this command to view the address family and routing protocol for an IP routing policy rule.
policy route pbr name <name> rule <number> description <description>

Provides a brief description for an IP routing policy rule.

**Syntax**

```
set policy route pbr name name rule number description description
delete policy route pbr name name rule number description
show policy route pbr name name rule number description
```

**Parameters**

- `name`
  - The name of an IP routing policy.
- `number`
  - The numeric identifier of the rule. The numbers range from 1 through 9999.
- `description`
  - A brief description for the rule. If the description contains spaces, it must be enclosed in double quotation marks (").

**Modes**

Configuration mode

**Configuration Statement**

```
policy {
  route {
    pbr {
      name name {
        rule number {
          description description
        }
      }
    }
  }
}
```

**Usage Guidelines**

Use the `set` form of this command to provide a description for an IP routing policy rule.

Use the `delete` form of this command to remove a description for an IP routing policy rule.

Use the `show` form of this command to display a description for an IP routing policy rule.
policy route pbr name <name> rule <number> destination <destination>

Defines the destination address for an IP routing policy rule.

**Syntax**

```
set policy route pbr name name rule number destination { address address | port [ port-name | 1-65535 | start-end | port-group-name ] }
```

```
delete policy route pbr name name rule number destination [ address address | port [ name | 1-65535 | start-end | port-group-name ] ]
```

```
show policy route pbr name name rule number destination [ address | port ]
```

**Parameters**

- **name**
  - The name of an IP routing policy.

- **number**
  - The numeric identifier of a policy rule. Rule numbers determine the order in which rules are processed. Each rule must have a unique rule number. The number ranges from 1 through 9999.
  - You can define multiple rules by creating more than one rule configuration node.

- **address**
  - A destination address to match. Address formats are as follows:
    - `ip-address`: An IP address.
    - `ip-address/prefix`: An IPv4 network address, where 0.0.0.0/0 matches any network.
    - `ip-address-ip-address`: A range of contiguous IPv4 addresses; for example, 192.168.1.1-192.168.1.150.
    - `!ip-address`: All IPv4 addresses except the one specified.
    - `!ip-address/prefix`: All IPv4 network addresses except the one specified.
    - `!ip-address-ip-address`: All IP addresses except those in the specified range.
    - `[ port-name | 1-65535 | start-end | port-group-name ]`
      - Applicable only when the protocol is TCP or UDP. A destination port to match. The format of the port is any of the following:
        - `port-name`: The name of an IP service; for example, http. You can specify any service name in the /etc/services file.
        - `1-65535`: A port number. The numbers range from 1 through 65535.
        - `start-end`: A specified range of ports; for example, 1001-1005.
        - `port-group-name`: A port group. A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

This criterion specifies a group of addresses, ports, or networks for packet destination address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the
packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the destination of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group, and a network group may be specified with a port group. You cannot specify both an address and a network group.

You can use a combination of these formats in a list separated by commas. You can also negate the entire list by prefixing it with an exclamation mark (!); for example, !22,telnet,http,123,1001-1005.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Modes

Configuration mode

Configuration Statement

```plaintext
policy {
    route {
        pbr {
            name name {
                rule number {
                    destination {
                        address address
                        port port-name
                        port 1-65535
                        port start-end
                        port port-group-name
                    }
                }
            }
        }
    }
}
```

Usage Guidelines

Use the `set` form of this command to create or modify a rule within an IP routing policy.

Use the `delete` form of this command to remove a rule from an IP routing policy.

Use the `show` form of this command to display a rule within an IP routing policy.
policy route pbr name <name> rule <number> disable

Disables a routing policy rule.

Syntax

- `set policy route pbr name name rule number disable`
- `delete policy route pbr name name rule number disable`
- `show policy route pbr name name rule number`

Command Default

The rule is enabled.

Parameters

- `name` The name of an IP routing policy.
- `number` The numeric identifier of the rule. The numbers range from 1 through 9999.

Modes

- Configuration mode

Configuration Statement

```
policy {
    route {
        pbr {
            name name {
                rule number {
                    disable
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to disable a routing policy rule. Disabling a rule is a useful way to test how the policy route performs without a specific rule and without having to delete and reconfigure the rule.

Use the `set` form of this command to disable a routing policy rule.

Use the `delete` form of this command to re-enable a rule.

Use the `show` form of this command to display a routing policy rule.
policy route pbr name <name> rule <number> icmp <icmp>

Creates a routing policy rule to match Internet Control Message Protocol (ICMP) packets.

**Syntax**

```plaintext
set policy route pbr name name rule number icmp { code number | type number | type-name [ name ] }

delete policy route pbr name name rule number icmp [ code | type type | type-name [ name ] ]

show policy route pbr name name rule number icmp
```

**Parameters**

- **name**
  - The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

- **number**
  - The numeric identifier of the rule. The identifier ranges from 1 through 9999.

- **code number**
  - An IPv4 ICMP code (0 through 255).

- **type number**
  - An IPv4 ICMP type (0 through 255).

- **type-name name**
  - The name of an ICMP.

  For type options, refer to **ICMP Types** on page 41.

**Modes**

Configuration mode

**Configuration Statement**

```
policy {
  route {
    name name {
      rule number {
        icmp {
          code number
          type number
          type-name all
          type-name name
        }
      }
    }
  }
}
```

**Usage Guidelines**

Use the **set** form of this command to create a rule to match ICMP packets.

Use the **delete** form of this command to delete a rule that matches ICMP packets.

Use the **show** form of this command to display a rule that matches ICMP packets.
policy route pbr name <name> rule <number> icmpv6 <icmpv6>

policy route pbr name <name> rule <number> icmpv6 <icmpv6>

Creates a routing policy rule to match Internet Control Message Protocol (ICMP) IP packets for a routing policy rule.

Syntax

set policy route pbr name name rule number icmpv6 [ code number ] [ type number ] [ type-name [ any | name ] ]
delete policy route pbr name name rule number icmpv6 [ code number ] [ type number ] [ type-name [ any | name ] ]
show policy route pbr name name rule number icmpv6

Command Default

The rule is enabled.

Parameters

name

The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

number

The numeric identifier of the rule. The identifier ranges from 1 through 9999.

code number

An IPv6 ICMP code (0 through 255).

type number

An IPv6 ICMP type (0 through 255).

A valid ICMPv6 type code from 0 through 255; for example, 128 (Echo Request), or a type and code pair (each from 0 to 255); for example, 1/4 for port unreachable. Alternatively, you can specify an ICMPv6 type code explicitly; for example, echo-request (Echo Request).

For a list of ICMP codes and types, see ICMPv6 Types on page 43.

type-name [ any | name ]

The name of an ICMPv6.

any: Any ICMPv6 type.

name: For type options, refer to ICMPv6 Types on page 43.

Modes

Configuration mode

Configuration Statement

policy {
  route {
    pbr {
      name name {
        rule number {
          icmpv6
            code number
            type number
            type-name any
            type-name name
          }
        }
      }
    }
  }
}

Usage Guidelines

Use the set form of this command to create a rule to match ICMP packets.

Use the delete form of this command to delete a rule that matches ICMP packets.

Use the show form of this command to view a rule that matches ICMP packets.
policy route pbr name <name> rule <number> log

Enables logging for a routing policy rule.

Syntax

set policy route pbr name name rule number log

delete policy route pbr name name rule number log

show policy route pbr name name rule number

Command Default
Logging is disabled.

Parameters

name
The name of an IP routing policy.

number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one rule configuration node.

Modes
Configuration mode

Configuration Statement

policy {
  route {
    pbr {
      name name {
        rule number {
          log
        }
      }
    }
  }
}

Usage Guidelines
When logging is enabled, any action taken is logged.

Use the set form of this command to enable logging for a routing policy rule.

Use the delete form of this command to restore the default behavior for logging, that is, actions are not logged.

Use the show form of this command to display whether logging is enabled or disabled.
policy route pbr name <name> rule <number> port <port-name>

Defines the source port name, number, range, or port group for a routing policy rule.

**Syntax**

```plaintext
set policy route pbr name rule number { port [ name | 1-65535 | start-end | port-group-name ] }

delete policy route pbr name rule number [ port [ name | 1-65535 | start-end | port-group-name ] ]

show policy route pbr name rule number [ port ]
```

**Parameters**

- **name**
  - The name of an IP routing policy.

- **port [ name | 1-65535 | start-end | port-group-name ]**
  - Applicable only when the protocol is TCP or UDP. A source port to match. The format of the port is any of the following:
    - **port-name**: The name of an IP service; for example, http. You can specify any service name in the `/etc/services` file.
    - **1-65535**: A port number. The numbers range from 1 through 65535.
    - **start-end**: A specified range of ports; for example, 1001-1005.
    - **port-group-name**: A port group. A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

This criterion specifies a group of addresses, ports, or networks for packet source address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the source of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group, and a network group may be specified with a port group. You cannot specify both an address and a network group.

You can use a combination of these formats in a list separated by commas. You can also negate the entire list by prefixing it with an exclamation mark (!); for example, !22,telnet,http,123,1001-1005.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

**Modes**

- Configuration mode

```plaintext
policy {
  route {
    pbr {
      name name {
        rule number {
          port name
          port 1-65535
          port start-end
          port port-group-name
        }
      }
    }
  }
}
```
**Usage Guidelines**

This criterion specifies a port or a group of ports for packet source address for a routing policy rule.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's source must match at least one item in the address group and at least one item in the port group.

An address group can be specified together with a port group, and a network group can be specified together with a port group. You cannot specify both an address and a network group.

The address family must match the specified family by using the `set policy route pbr name name rule number address-family ipv4` command.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.
Defines the protocol of an IP routing policy rule.

**Syntax**

```plaintext
set policy route pbr name name rule number protocol { text | 0-255 | all | name }
delete policy route pbr name name rule number protocol [ text | 0-255 | all | name ]
show policy route pbr name name rule number protocol
```

**Parameters**

- **name**
  - The name of an IP routing policy.

- **number**
  - The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.
  - You can define multiple rules by creating more than one `rule` configuration node.

- **protocol**
  - The protocol is any of the following:
    - **text:** Matches packets by protocol type. Any protocol literals or numbers listed in the file `/etc/protocols` can be specified. The keywords `icmpv6` and `all` (for all protocols) are also supported.
    - **0-255:** An IP protocol number that ranges from 0 through 255.
    - **all:** All IP protocols.
    - **! protocol:** All IP protocols except for the specified name or number. Prefixing the protocol name with the negation operator (the exclamation mark) matches every protocol except the specified protocol. For example, `!tcp` matches all protocols except TCP.

  This parameter matches the last, next-header field in the IP header chain. This match means that if the packet has no extension headers, it matches the next-header field in the main header. If the packet does have extension headers, the parameter matches the next-header field of the last extension header in the chain. In other words, the parameter always matches the ID of the transport-layer packet that is being carried.

  Exercise care when employing more than one rule that uses the negation. Routing policy rules are evaluated sequentially, and a sequence of negated rules could result in unexpected behavior.

**Modes**

<table>
<thead>
<tr>
<th>Configuration Statement</th>
<th>Configuration mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy {</td>
<td></td>
</tr>
<tr>
<td>route {</td>
<td></td>
</tr>
<tr>
<td>pbr {</td>
<td></td>
</tr>
<tr>
<td>name name {</td>
<td></td>
</tr>
<tr>
<td>rule number {</td>
<td></td>
</tr>
<tr>
<td>protocol</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td></td>
</tr>
<tr>
<td>0-255</td>
<td></td>
</tr>
<tr>
<td>all</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
Usage Guidelines

Use the **set** form of this command to define the protocol of an IP routing policy rule.
Use the **delete** form of this command to remove a protocol from a routing policy rule.
Use the **show** form of this command to view the protocol of a routing policy rule.
policy route pbr name <name> rule <number> set <match-criteria>

Defines the address family or routing table ID for an IP routing policy rule.

Syntax

set policy route pbr name name rule rule-number set { af name | table number }
dele policy route pbr name name rule rule-number set { af name | table number }
show policy route pbr name name rule rule-number set

Parameters

name
The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one rule configuration node.

af name
To match IPv4 or IPv6 address family for this rule. Performs alternate processing on packets satisfying the match criteria.

table number
To match according to the PBR Table ID numbers 1 through 128. Performs alternate processing on packets satisfying the match criteria.

Modes

Configuration mode

Configuration Statement

policy {
  route {
    pbr {
      name name {
        rule number {
          set
            af name
            table number
        }
      }
    }
  }
}

Usage Guidelines

Use the set form of this command to define the address family or routing table ID for an IP routing policy rule.

Use the delete form of this command to remove the address family or routing table ID for a rule.

Use the show form of this command to view the address family or routing table ID for a rule.

The address family must match the specified family by using the set policy route pbr name name rule number address-family ipv4 command.

Use the set form of this command to define the source for a routing policy rule.

Use the delete form of this command to remove the source for a routing policy rule.

Use the show form of this command to view the source for a routing policy rule.
policy route pbr name <name> rule <number> source address <address>

Defines the source address for a routing policy rule.

Syntax

set policy route pbr name name rule number source address address
delete policy route pbr name name rule number source address address
show policy route pbr name name rule number source address address

Parameters

name
The name of an IP routing policy.

address address
A source address to match. The match criteria are any of the following:

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.
ipv6-address/prefix: A network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.
ipv6-address-ipv6-address: A range of contiguous IP addresses; for example, fe80::20c:29fe:fe47:f00-fe80::20c:29fe:fe47:f89.
!ipv6-address: All IP addresses except the one specified.
!ipv6-address/prefix: All network addresses except the one specified.
!ipv6-address-ipv6-address: All IP addresses except those in the specified range.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Modes

Configuration mode

Usage Guidelines

This criterion specifies a port or a group of ports for packet source address for a routing policy rule.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's source must match at least one item in the address group and at least one item in the port group.

An address group can be specified together with a port group, and a network group can be specified together with a port group. You cannot specify both an address and a network group.
The address family must match the specified family by using the `set policy route pbr name name rule number address-family ipv4` command.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.
policy route pbr name <name> rule <number> source port <port>

Defines the source port name, number, range, or port group for a routing policy rule.

Syntax

```
set policy route pbr name name rule number source port [ name | 1-65535 | start-end | port-group-name ]
```

```
delete policy route pbr name name rule number source port [ name | 1-65535 | start-end | port-group-name ]
```

```
show policy route pbr name name rule number source port
```

Parameters

- **name**
  - The name of an IP routing policy.

- **port**
  - `name` | `1-65535` | `start-end` | `port-group-name`
  - Applicable only when the protocol is TCP or UDP. A source port to match. The format of the port is any of the following:
    - `name`: The name of an IP service; for example, `http`. You can specify any service name in the `/etc/services` file.
    - `1-65535`: A port number. The numbers range from 1 through 65535.
    - `start-end`: A specified range of ports; for example, `1001-1005`.
    - `port-group-name`: A port group. A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

This criterion specifies a group of addresses, ports, or networks for packet source address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the source of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group, and a network group may be specified with a port group. You cannot specify both an address and a network group.

You can use a combination of these formats in a list separated by commas. You can also negate the entire list by prefixing it with an exclamation mark (!); for example, `!22,telnet,http,123,1001-1005`.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.
Usage Guidelines

This criterion specifies a port or a group of ports for packet source address for a routing policy rule.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's source must match at least one item in the address group and at least one item in the port group.

Use the set form of this command to define the source for a routing policy rule.

Use the delete form of this command to remove the source for a routing policy rule.

Use the show form of this command to view the source for a routing policy rule.
policy route pbr name <name> rule <number> table

Defines the table number for an IP routing policy rule.

**Syntax**

```plaintext
set policy route pbr name name rule rule-number table number
delete policy route pbr name name rule rule-number table [ number ]
show policy route pbr name name rule rule-number set
```

**Parameters**

- **name**
  - The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

- **number**
  - The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

  You can define multiple rules by creating more than one `rule` configuration node.

- **af name**
  - To match IPv4 or IPv6 address family for this rule. Performs alternate processing on packets satisfying the match criteria.

- **table number**
  - To match according to the PBR Table ID numbers 1 through 128. Performs alternate processing on packets satisfying the match criteria.

**Modes**

- **Configuration mode**

**Configuration Statement**

```plaintext
policy {
  route {
    pbr {
      name name {
        rule number {
          set af name
          table number
        }
        } rule-number
    }
  }
}
```

**Usage Guidelines**

- Use the `set` form of this command to define the address family or routing table ID for an IP routing policy rule.
- Use the `delete` form of this command to remove the address family or routing table ID for a rule.
- Use the `show` form of this command to view the address family or routing table ID for a rule.
- The address family must match the specified family by using the `set policy route pbr name name rule number address-family ipv4` command.
- Use the `set` form of this command to define the source for a routing policy rule.
- Use the `delete` form of this command to remove the source for a routing policy rule.
- Use the `show` form of this command to view the source for a routing policy rule.
policy route pbr name <name> rule <number> tcp flags <flags>

Defines the types of TCP flags to be matched for a routing policy rule.

Syntax

set policy route pbr name name rule number tcp flags flags
delete policy route pbr name name rule number tcp flags [ flags ]
show policy route pbr name name rule number tcp flags

Parameters

name
The name of an IP routing policy.

number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one rule configuration node.

flags
The flags to be matched in a packet. The flags are any of SYN, ACK, FIN, RST, URG, and PSH. You can specify more than one flag in a list separated by commas.

Prefixing a flag name with the negation operator matches packets with that flag unset. You can also use ! to match packets by not using a given TCP flag. For example, the list SYN, !ACK, !FIN, !RST matches only packets with the SYN flag set and the ACK, FIN, and RST flags unset.

Modes

Configuration mode

Usage Guidelines

Use the set form of this command to define the types of TCP flags to be matched for a routing policy rule.

Use the delete form of this command to remove the types of TCP flags to be matched for a routing policy rule.

Use the show form of this command to view the types of TCP flags to be matched for a routing policy rule.
show policy route <interface>

Displays routing policy configuration or statistics.

**Syntax**

```
show policy route interface
```

**Parameters**

`interface`

The name of an interface.

**Modes**

Operational mode

**Usage Guidelines**

A policy identifies traffic that matches parameters and specifies which table to use. The table defines the routes for a packet to take. A routing policy is a named collection of as many as 9,999 packet-classification rules. When applied to an interface, the policy rule classifies incoming traffic.

**NOTE**

The PBR rule counters count all of the matched packets regardless of the availability of the route.

Use this command in operational mode to display packet statistics for all PBR rules in all groups.

For example:

```
show policy route
```

```
vytta@vyatta:~$ show policy route
----------------------------------------------------------
Rulesets Information
----------------------------------------------------------
PBR Group: "dp0p192p1-group-v6":
Active on (dp0p192p1, in) 
rule  proto  packets  bytes
----  -----  -------  ----- 
1    icmpv6  0 0
condition - proto icmpv6  from abcd::1 to 5555::1 tag 98
2    tcp  0 0
condition - proto tcp from abcd::1 to 5555::1 tag 99
3    udp  0 0
condition - proto udp from abcd::1 to 3333::1 tag 100
```

**Related commands**

The following table lists related commands that are documented elsewhere.

<table>
<thead>
<tr>
<th>Related commands documented elsewhere</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocols static table</td>
</tr>
<tr>
<td>resources group address-group</td>
</tr>
<tr>
<td>&lt;group-name&gt;</td>
</tr>
</tbody>
</table>

Policy-Based Routing Reference Guide
53-1003723-02
### Related commands documented elsewhere

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources group port-group &lt;group-name&gt;</td>
<td>Defines a group of ports that are referenced in firewall rules. (Refer to <em>Basic Routing Reference Guide</em>.)</td>
</tr>
<tr>
<td>show ip route table</td>
<td>The command for displaying the contents of an alternate routing table is described in <em>Vyatta Basic Routing Reference Guide</em>.</td>
</tr>
<tr>
<td>firewall group</td>
<td>Routing policy match criteria support references to predefined groups of addresses, ports, and networks. Commands for defining such groups are described in <em>Vyatta Firewall Reference Guide</em>.</td>
</tr>
</tbody>
</table>
This appendix lists the Internet Control Messaging Protocol (ICMP) types defined by the Internet Assigned Numbers Authority (IANA).

The IANA has developed a standard that maps a set of integers onto ICMP types. The following table lists the ICMP types and codes defined by the IANA and maps them to the literal strings that are available in the Vyatta system.

<table>
<thead>
<tr>
<th>ICMP Type</th>
<th>Code</th>
<th>Literal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Echo reply</td>
<td>0</td>
<td>echo-reply</td>
<td>Echo reply (pong)</td>
</tr>
<tr>
<td>3 - Destination unreachable</td>
<td></td>
<td>destination- unreachable</td>
<td>Destination is unreachable</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>network-unreachable</td>
<td>Destination network is unreachable</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>host-unreachable</td>
<td>Destination host is unreachable</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>protocol-unreachable</td>
<td>Destination protocol is unreachable</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>port-unreachable</td>
<td>Destination port is unreachable</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>fragmentation-needed</td>
<td>Fragmentation is required</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>source-route-failed</td>
<td>Source route has failed</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>network-unknown</td>
<td>Destination network is unknown</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>host-unknown</td>
<td>Destination host is unknown</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>network-prohibited</td>
<td>Network is administratively prohibited</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>host-prohibited</td>
<td>Host is administratively is prohibited</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>ToS-network-unreachable</td>
<td>Network is unreachable for ToS</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>ToS-host-unreachable</td>
<td>Host is unreachable for ToS</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>communication-prohibited</td>
<td>Communication is administratively prohibited</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>host-precedence-violation</td>
<td>Requested precedence is not permitted.</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>precedence-cutoff</td>
<td>Precedence is lower than the required minimum.</td>
</tr>
<tr>
<td>4 - Source quench</td>
<td>0</td>
<td>source-quench</td>
<td>Source is quenched (congestion control)</td>
</tr>
<tr>
<td>5 - Redirect message</td>
<td></td>
<td>redirect</td>
<td>Redirected message</td>
</tr>
</tbody>
</table>
TABLE 3  ICMP types (Continued)

<table>
<thead>
<tr>
<th>ICMP Type</th>
<th>Code</th>
<th>Literal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 network-redirect</td>
<td>0</td>
<td>network-redirect</td>
<td>Datagram is redirected for the network</td>
</tr>
<tr>
<td>1 host-redirect</td>
<td>1</td>
<td>host-redirect</td>
<td>Datagram is redirected for the host</td>
</tr>
<tr>
<td>2 ToS-network-redirect</td>
<td>2</td>
<td>ToS-network-redirect</td>
<td>Datagram is redirected for the ToS and network</td>
</tr>
<tr>
<td>3 ToS-host-redirect</td>
<td>3</td>
<td>ToS-host-redirect</td>
<td>Datagram is redirected for the ToS and host</td>
</tr>
<tr>
<td>8 - Echo request</td>
<td>0</td>
<td>echo-request</td>
<td>Echo request (ping)</td>
</tr>
<tr>
<td>9 - Router advertisement</td>
<td>0</td>
<td>router-advertisement</td>
<td>Router advertisement</td>
</tr>
<tr>
<td>10 - Router solicitation</td>
<td>0</td>
<td>router-solicitation</td>
<td>Router solicitation</td>
</tr>
<tr>
<td>11 - Time exceeded</td>
<td>time-exceeded</td>
<td>time-exceeded</td>
<td>Time to live (TTL) has exceeded</td>
</tr>
<tr>
<td>8 - Echo request</td>
<td>0</td>
<td>echo-request</td>
<td>Echo request (ping)</td>
</tr>
<tr>
<td>9 - Router advertisement</td>
<td>0</td>
<td>router-advertisement</td>
<td>Router advertisement</td>
</tr>
<tr>
<td>10 - Router solicitation</td>
<td>0</td>
<td>router-solicitation</td>
<td>Router solicitation</td>
</tr>
<tr>
<td>11 - Time exceeded</td>
<td>time-exceeded</td>
<td>time-exceeded</td>
<td>Time to live (TTL) has exceeded</td>
</tr>
<tr>
<td>12 - Parameter problem: Bad IP header</td>
<td>parameter-problem</td>
<td>parameter-problem</td>
<td>Bad IP header</td>
</tr>
<tr>
<td>0 ip-header-bad</td>
<td>0</td>
<td>ip-header-bad</td>
<td>Pointer that indicates an error</td>
</tr>
<tr>
<td>1 required-option-missing</td>
<td>1</td>
<td>required-option-missing</td>
<td>Missing required option</td>
</tr>
<tr>
<td>13 - Timestamp</td>
<td>timestamp-request</td>
<td>timestamp-request</td>
<td>Request for a timestamp</td>
</tr>
<tr>
<td>14 - Timestamp reply</td>
<td>timestamp-reply</td>
<td>timestamp-reply</td>
<td>Reply to a request for a timestamp</td>
</tr>
<tr>
<td>15 - Information request</td>
<td>0</td>
<td>Information request</td>
<td>Information request</td>
</tr>
<tr>
<td>16 - Information reply</td>
<td>0</td>
<td>Information reply</td>
<td>Information reply</td>
</tr>
<tr>
<td>17 - Address mask request</td>
<td>address-mask-request</td>
<td>address-mask-request</td>
<td>Address mask request</td>
</tr>
<tr>
<td>18 - Address mask reply</td>
<td>address-mask-reply</td>
<td>address-mask-reply</td>
<td>Address mask reply</td>
</tr>
<tr>
<td>19 - Ping</td>
<td>ping</td>
<td>ping</td>
<td>A ping message</td>
</tr>
<tr>
<td>20 - Pong</td>
<td>pong</td>
<td>pong</td>
<td>A pong message</td>
</tr>
</tbody>
</table>
ICMPv6 Types

This appendix lists the ICMPv6 types defined by the Internet Assigned Numbers Authority (IANA).

The Internet Assigned Numbers Authority (IANA) has developed a standard that maps a set of integers onto ICMPv6 types. The following table lists the ICMPv6 types and codes defined by the IANA and maps them to the strings literal strings available in the Vyatta system.

**TABLE 4  ICMPv6 types**

<table>
<thead>
<tr>
<th>ICMPv6 Type</th>
<th>Code</th>
<th>Literal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Destination unreachable</td>
<td>0</td>
<td>no-route</td>
<td>No route to destination</td>
</tr>
<tr>
<td>1 - Destination unreachable</td>
<td>1</td>
<td>communication-prohibited</td>
<td>Communication with destination administratively prohibited</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>Beyond scope of source address</td>
</tr>
<tr>
<td>2 - Packet too big</td>
<td>0</td>
<td>packet-too-big</td>
<td></td>
</tr>
<tr>
<td>3 - Time exceeded</td>
<td>0</td>
<td>ttl-zero-during-transit</td>
<td>Hop limit exceeded in transit</td>
</tr>
<tr>
<td>3 - Time exceeded</td>
<td>1</td>
<td>ttl-zero-during-reassembly</td>
<td>Fragment reassembly time exceeded</td>
</tr>
<tr>
<td>4 - Parameter problem</td>
<td>0</td>
<td>bad-header</td>
<td>Erroneous header field encountered</td>
</tr>
<tr>
<td>4 - Parameter problem</td>
<td>1</td>
<td>unknown-header-type</td>
<td>Unrecognized Next Header type encountered</td>
</tr>
<tr>
<td>4 - Parameter problem</td>
<td>2</td>
<td>unknown-option</td>
<td>Unrecognized IPv6 option encountered</td>
</tr>
<tr>
<td>128 - Echo request</td>
<td>0</td>
<td>echo-request (ping)</td>
<td>Echo request</td>
</tr>
<tr>
<td>129 - Echo reply</td>
<td>0</td>
<td>echo-reply (pong)</td>
<td>Echo reply</td>
</tr>
<tr>
<td>133 - Router solicitation</td>
<td>0</td>
<td>router-solicitation</td>
<td>Router solicitation</td>
</tr>
<tr>
<td>ICMPv6 Type</td>
<td>Code</td>
<td>Literal</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------</td>
<td>---------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>134 - Router advertisement</td>
<td>0</td>
<td>router-advertisement</td>
<td>Router advertisement</td>
</tr>
<tr>
<td>135 - Neighbor solicitation</td>
<td>0</td>
<td>neighbor-solicitation (neighbour-solicitation)</td>
<td>Neighbor solicitation</td>
</tr>
<tr>
<td>136 - Neighbor advertisement</td>
<td>0</td>
<td>neighbor-advertisement (neighbour-advertisement)</td>
<td>Neighbor advertisement</td>
</tr>
</tbody>
</table>
**Supported Interface Types**

The following table shows the syntax and parameters of supported interface types. Depending on the command, some of these types may not apply.

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Syntax</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge</td>
<td>bridge brx</td>
<td>brx: The name of a bridge group. The name ranges from br0 through br999.</td>
</tr>
<tr>
<td>Dataplane</td>
<td>dataplane interface-name</td>
<td>interface-name: The name of a dataplane interface. Following are the supported formats of the interface name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dpxpypz — The name of a dataplane interface, where</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— dp specifies the dataplane identifier (ID). Currently, only dp0 is supported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— py specifies a physical or virtual PCI slot index (for example, p129).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— pz specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dpxem — The name of a dataplane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where emy specifies an embedded network interface number (typically, a small number). For example, dp0em3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dpxsy — The name of a dataplane interface on a device that is installed on a virtual PCI slot, where xsy specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dpPnypypz — The name of a dataplane interface on a device that is installed on a secondary PCI bus, where Pn specifies the bus number. You can use this format to name dataplane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of n must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.</td>
</tr>
<tr>
<td>Dataplane vif</td>
<td>dataplane interface-name vif-id [vlan vlan-id ]</td>
<td>interface-name: Refer to the preceding description.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vif-id: A virtual interface ID. The ID ranges from 1 through 4094.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vlan-id: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.</td>
</tr>
<tr>
<td>Loopback</td>
<td>loopback lo</td>
<td>n: The name of a loopback interface, where n ranges from 1 through 99999.</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td>loopback loni</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>openvpn vtunx</td>
<td>vtunx: The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtunx, where x is a nonnegative integer.</td>
</tr>
</tbody>
</table>
### Supported Interface Types

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Syntax</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel</td>
<td><strong>tunnel</strong> tunx</td>
<td><em>tunx</em>: The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>tunnel</strong> tunx</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Virtual tunnel</td>
<td><strong>vti</strong> vtxn</td>
<td><em>vtxn</em>: The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vtxn, where x is a nonnegative integer.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This interface does not support IPv6.</td>
<td></td>
</tr>
<tr>
<td>VRRP</td>
<td><strong>parent-interface vrrp</strong></td>
<td><em>parent-interface</em>: The type and identifier of a parent interface; for example, dataplane dp0p1p2 or bridge br999.</td>
</tr>
<tr>
<td></td>
<td><strong>vrrp-group group</strong></td>
<td><em>group</em>: A VRRP group identifier.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as does the parent interface.</td>
</tr>
</tbody>
</table>
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>access control list</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>AH</td>
<td>Authentication Header</td>
</tr>
<tr>
<td>AMI</td>
<td>Amazon Machine Image</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AS</td>
<td>autonomous system</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>AWS</td>
<td>Amazon Web Services</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>BIOS</td>
<td>Basic Input Output System</td>
</tr>
<tr>
<td>BPDU</td>
<td>Bridge Protocol Data Unit</td>
</tr>
<tr>
<td>CA</td>
<td>certificate authority</td>
</tr>
<tr>
<td>CCMP</td>
<td>AES in counter mode with CBC-MAC</td>
</tr>
<tr>
<td>CHAP</td>
<td>Challenge Handshake Authentication Protocol</td>
</tr>
<tr>
<td>CLI</td>
<td>command-line interface</td>
</tr>
<tr>
<td>DDNS</td>
<td>dynamic DNS</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DHCPv6</td>
<td>Dynamic Host Configuration Protocol version 6</td>
</tr>
<tr>
<td>DLCI</td>
<td>data-link connection identifier</td>
</tr>
<tr>
<td>DMI</td>
<td>desktop management interface</td>
</tr>
<tr>
<td>DMVPN</td>
<td>dynamic multipoint VPN</td>
</tr>
<tr>
<td>DMZ</td>
<td>demilitarized zone</td>
</tr>
<tr>
<td>DN</td>
<td>distinguished name</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>eBGP</td>
<td>external BGP</td>
</tr>
<tr>
<td>EBS</td>
<td>Amazon Elastic Block Storage</td>
</tr>
<tr>
<td>EC2</td>
<td>Amazon Elastic Compute Cloud</td>
</tr>
<tr>
<td>EGP</td>
<td>Exterior Gateway Protocol</td>
</tr>
<tr>
<td>ECMP</td>
<td>equal-cost multipath</td>
</tr>
<tr>
<td>ESP</td>
<td>Encapsulating Security Payload</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>FIB</td>
<td>Forwarding Information Base</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GRE</td>
<td>Generic Routing Encapsulation</td>
</tr>
<tr>
<td>HDLC</td>
<td>High-Level Data Link Control</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IGMP</td>
<td>Internet Group Management Protocol</td>
</tr>
<tr>
<td>IGP</td>
<td>Interior Gateway Protocol</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Protection System</td>
</tr>
<tr>
<td>IKE</td>
<td>Internet Key Exchange</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPOA</td>
<td>IP over ATM</td>
</tr>
<tr>
<td>IPsec</td>
<td>IP Security</td>
</tr>
<tr>
<td>IPv4</td>
<td>IP Version 4</td>
</tr>
<tr>
<td>IPv6</td>
<td>IP Version 6</td>
</tr>
<tr>
<td>ISAKMP</td>
<td>Internet Security Association and Key Management Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Internet Standard Multicast</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>KVM</td>
<td>Kernel-Based Virtual Machine</td>
</tr>
<tr>
<td>L2TP</td>
<td>Layer 2 Tunneling Protocol</td>
</tr>
<tr>
<td>LACP</td>
<td>Link Aggregation Control Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>local area network</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LLDP</td>
<td>Link Layer Discovery Protocol</td>
</tr>
<tr>
<td>MAC</td>
<td>medium access control</td>
</tr>
<tr>
<td>mGRE</td>
<td>multipoint GRE</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base</td>
</tr>
<tr>
<td>MLD</td>
<td>Multicast Listener Discovery</td>
</tr>
<tr>
<td>MLP PPP</td>
<td>multilink PPP</td>
</tr>
<tr>
<td>MRRU</td>
<td>maximum received reconstructed unit</td>
</tr>
<tr>
<td>MTU</td>
<td>maximum transmission unit</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>NBMA</td>
<td>Non-Broadcast Multi-Access</td>
</tr>
<tr>
<td>ND</td>
<td>Neighbor Discovery</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>NHRP</td>
<td>Next Hop Resolution Protocol</td>
</tr>
<tr>
<td>NIC</td>
<td>network interface card</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>OSPFv2</td>
<td>OSPF Version 2</td>
</tr>
<tr>
<td>OSPFv3</td>
<td>OSPF Version 3</td>
</tr>
<tr>
<td>PAM</td>
<td>Pluggable Authentication Module</td>
</tr>
<tr>
<td>PAP</td>
<td>Password Authentication Protocol</td>
</tr>
<tr>
<td>PAT</td>
<td>Port Address Translation</td>
</tr>
<tr>
<td>PCI</td>
<td>peripheral component interconnect</td>
</tr>
<tr>
<td>PIM</td>
<td>Protocol Independent Multicast</td>
</tr>
<tr>
<td>PIM-DM</td>
<td>PIM Dense Mode</td>
</tr>
<tr>
<td>PIM-SM</td>
<td>PIM Sparse Mode</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-Point Protocol</td>
</tr>
<tr>
<td>PPPoA</td>
<td>PPP over ATM</td>
</tr>
<tr>
<td>PPPoE</td>
<td>PPP over Ethernet</td>
</tr>
<tr>
<td>PPTP</td>
<td>Point-to-Point Tunneling Protocol</td>
</tr>
<tr>
<td>PTMU</td>
<td>Path Maximum Transfer Unit</td>
</tr>
<tr>
<td>PVC</td>
<td>permanent virtual circuit</td>
</tr>
<tr>
<td>QoS</td>
<td>quality of service</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Authentication Dial-In User Service</td>
</tr>
<tr>
<td>RHEL</td>
<td>Red Hat Enterprise Linux</td>
</tr>
<tr>
<td>RIB</td>
<td>Routing Information Base</td>
</tr>
<tr>
<td>RIP</td>
<td>Routing Information Protocol</td>
</tr>
<tr>
<td>RIPng</td>
<td>RIP next generation</td>
</tr>
<tr>
<td>RP</td>
<td>Rendezvous Point</td>
</tr>
<tr>
<td>RPF</td>
<td>Reverse Path Forwarding</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest, Shamir, and Adleman</td>
</tr>
<tr>
<td>Rx</td>
<td>receive</td>
</tr>
<tr>
<td>S3</td>
<td>Amazon Simple Storage Service</td>
</tr>
<tr>
<td>SLAAC</td>
<td>Stateless Address Auto-Configuration</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SONET</td>
<td>Synchronous Optical Network</td>
</tr>
<tr>
<td>SPT</td>
<td>Shortest Path Tree</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSID</td>
<td>Service Set Identifier</td>
</tr>
<tr>
<td>SSM</td>
<td>Source-Specific Multicast</td>
</tr>
<tr>
<td>STP</td>
<td>Spanning Tree Protocol</td>
</tr>
<tr>
<td>TACACS+</td>
<td>Terminal Access Controller Access Control System Plus</td>
</tr>
<tr>
<td>TBF</td>
<td>Token Bucket Filter</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TKIP</td>
<td>Temporal Key Integrity Protocol</td>
</tr>
<tr>
<td>ToS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>TSS</td>
<td>TCP Maximum Segment Size</td>
</tr>
<tr>
<td>Tx</td>
<td>transmit</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>VHD</td>
<td>virtual hard disk</td>
</tr>
<tr>
<td>vif</td>
<td>virtual interface</td>
</tr>
<tr>
<td>VLAN</td>
<td>virtual LAN</td>
</tr>
<tr>
<td>VPC</td>
<td>Amazon virtual private cloud</td>
</tr>
<tr>
<td>VPN</td>
<td>virtual private network</td>
</tr>
<tr>
<td>VRRP</td>
<td>Virtual Router Redundancy Protocol</td>
</tr>
<tr>
<td>WAN</td>
<td>wide area network</td>
</tr>
<tr>
<td>WAP</td>
<td>wireless access point</td>
</tr>
<tr>
<td>WPA</td>
<td>Wired Protected Access</td>
</tr>
</tbody>
</table>