



Junos[®] OS

RIP Configuration Guide

Release

13.1



Published: 2013-02-14

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® OS RIP Configuration Guide

13.1

Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Supported Platforms	xi
	Using the Examples in This Manual	xii
	Merging a Full Example	xii
	Merging a Snippet	xiii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xv
	Opening a Case with JTAC	xvi
Part 1	Overview	
Chapter 1	Introduction to RIP	3
	RIP Overview	3
	Distance-Vector Routing Protocols	3
	RIP Protocol Overview	4
	RIP Packets	5
	Maximizing Hop Count	6
	Split Horizon and Poison Reverse Efficiency Techniques	6
	Limitations of Unidirectional Connectivity	7
	RIP Configuration Overview	8
Chapter 2	RIP Standards	9
	Supported RIP and RIPng Standards	9
Part 2	Configuration	
Chapter 3	Concepts and Examples	13
	Example: Configuring RIP	13
	Understanding Basic RIP Routing	13
	Example: Configuring a Basic RIP Network	14
	Example: Configuring Authentication for RIP Routes	20
	Understanding RIP Authentication	20
	Example: Configuring Route Authentication for RIP	20
	Enabling Authentication with Plain-Text Passwords (CLI Procedure)	25
	Enabling Authentication with MD5 Authentication (CLI Procedure)	25
	Example: Configuring BFD for RIP	26
	Understanding BFD for RIP	26
	Example: Configuring BFD for RIP	27

Example: Configuring BFD Authentication for RIP	32
Understanding BFD Authentication for RIP	32
BFD Authentication Algorithms	33
Security Authentication Keychains	33
Strict Versus Loose Authentication	34
Example: Configuring BFD Authentication for RIP	34
Example: Configuring Point-to-Multipoint RIP Networks	40
Configuring Point-to-Multipoint RIP Networks Overview	40
Example: Configuring Point-to-Multipoint RIP Networks	41
Example: Applying Policies to RIP Routes Imported from Neighbors	47
Understanding RIP Import Policy	47
Example: Applying Policies to RIP Routes Imported from Neighbors	47
Examples: Controlling Traffic with Metrics in a RIP Network	52
Understanding Traffic Control with Metrics in a RIP Network	52
Example: Controlling Traffic in a RIP Network with an Incoming Metric	53
Example: Controlling Traffic in a RIP Network with an Outgoing Metric	55
Example: Configuring the Metric Value Added to Imported RIP Routes	56
Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets ..	60
Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets	61
Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets	61
Example: Redistributing Routes Among RIP Instances	64
Understanding Route Redistribution Among RIP instances	64
Example: Redistributing Routes Between Two RIP Instances	65
Example: Configuring RIP Timers	70
Understanding RIP Timers	70
Example: Configuring RIP Timers	71
Example: Configuring RIP Demand Circuits	77
RIP Demand Circuits Overview	77
RIP Demand Circuit Packets	78
Timers Used by RIP Demand Circuits	79
Example: Configuring RIP Demand Circuits	79
Example: Configuring Point-to-Multipoint RIP Networks	82
Configuring Point-to-Multipoint RIP Networks Overview	82
Example: Configuring Point-to-Multipoint RIP Networks	83
Example: Tracing RIP Protocol Traffic	89
Understanding RIP Trace Operations	89
Example: Tracing RIP Protocol Traffic	90
Verifying a RIP Configuration	94
Verifying the Exchange of RIP Messages	94
Verifying the RIP-Enabled Interfaces	96
Verifying Reachability of All Hosts in the RIP Network	96
Chapter 4 Configuration Statements	99
[edit protocols rip] Hierarchy Level	99
any-sender	101
authentication-key (Protocols RIP)	102
authentication-type (Protocols RIP)	103
bfd-liveness-detection (Protocols RIP)	104

check-zero	107
demand-circuit (Protocols RIP)	108
dynamic-peers	108
export (Protocols RIP)	109
graceful-restart (Protocols RIP)	110
group (Protocols RIP)	111
holddown (Protocols RIP)	113
import (Protocols RIP)	114
interface-type (Protocols RIP)	115
max-retrans-time	116
message-size	117
metric-in (Protocols RIP)	118
metric-out (Protocols RIP)	119
neighbor (Protocols RIP)	120
peer (Protocols RIP)	121
preference (Protocols RIP)	122
receive (Protocols RIP)	123
rib-group (Protocols RIP)	124
rip	124
route-timeout (Protocols RIP)	125
routing-instances (Multiple Routing Entities)	126
send (Protocols RIP)	127
traceoptions (Protocols RIP)	128
update-interval (Protocols RIP)	131

Part 3

Chapter 5

Administration

Operational Commands	135
clear rip general-statistics	136
clear rip statistics	137
restart	138
show policy	148
show policy conditions	150
show rip general-statistics	152
show rip neighbor	154
show rip statistics	156
show route	159
show route active-path	165
show route advertising-protocol	170
show route all	175
show route best	177
show route brief	180
show route detail	182
show route exact	197
show route export	199
show route extensive	201
show route forwarding-table	216
show route hidden	230
show route inactive-path	233

	show route instance	237
	show route next-hop	244
	show route output	250
	show route protocol	254
	show route receive-protocol	266
	show route table	275
	show route terse	286
	test policy	289
Part 4	Troubleshooting	
Chapter 6	Routing Protocol Process Memory FAQs	293
	Routing Protocol Process Memory FAQs Overview	293
	Routing Protocol Process Memory FAQs	294
	Frequently Asked Questions: Routing Protocol Process Memory	294
	Frequently Asked Questions: Interpreting Routing Protocol Process-Related Command Outputs	295
	Frequently Asked Questions: Routing Protocol Process Memory Swapping	298
	Frequently Asked Questions: Troubleshooting the Routing Protocol Process	299
Chapter 7	RIP Monitoring	301
	Monitoring RIP Routing Information	301
Part 5	Index	
	Index	305

List of Figures

Part 1	Overview	
Chapter 1	Introduction to RIP	3
	Figure 1: Distance-Vector Protocol	4
	Figure 2: Split Horizon Example	6
	Figure 3: Poison Reverse Example	7
	Figure 4: Limitations of Unidirectional Connectivity	7
Part 2	Configuration	
Chapter 3	Concepts and Examples	13
	Figure 5: Sample RIP Network Topology	14
	Figure 6: RIP Authentication Network Topology	21
	Figure 7: RIP BFD Network Topology	29
	Figure 8: RIP BFD Authentication Network Topology	35
	Figure 9: Configuring a Point-to-Multipoint RIP Network	43
	Figure 10: RIP Import Policy Network Topology	48
	Figure 11: Controlling Traffic in a RIP Network with the Incoming Metric	54
	Figure 12: Controlling Traffic in a RIP Network with the Outgoing Metric	55
	Figure 13: RIP Incoming Metrics Network Topology	57
	Figure 14: Sending and Receiving RIPv1 and RIPv2 Packets Network Topology ...	61
	Figure 15: Redistributing Routes Between RIP Instances Network Topology ...	66
	Figure 16: RIP Timers Network Topology	72
	Figure 17: Configuring a Point-to-Multipoint RIP Network	85
	Figure 18: RIP Trace Operations Network Topology	91

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiv
Part 2	Configuration	
Chapter 3	Concepts and Examples	13
	Table 3: Configuring Simple RIP Authentication	25
	Table 4: Configuring MD5 RIP Authentication	26
	Table 5: RIP Demand Circuit Packet Types	78
Part 3	Administration	
Chapter 5	Operational Commands	135
	Table 6: show policy Output Fields	148
	Table 7: show policy conditions Output Fields	150
	Table 8: show rip general-statistics Output Fields	152
	Table 9: show rip neighbor Output Fields	154
	Table 10: show rip statistics Output Fields	157
	Table 11: show route Output Fields	160
	Table 12: show route advertising-protocol Output Fields	171
	Table 13: show route detail Output Fields	182
	Table 14: Next-hop Types Output Field Values	186
	Table 15: State Output Field Values	188
	Table 16: Communities Output Field Values	190
	Table 17: show route export Output Fields	199
	Table 18: show route extensive Output Fields	201
	Table 19: show route forwarding-table Output Fields	218
	Table 20: show route instance Output Fields	237
	Table 21: show route receive-protocol Output Fields	266
	Table 22: show route terse Output Fields	286
Part 4	Troubleshooting	
Chapter 6	Routing Protocol Process Memory FAQs	293
	Table 23: show system processes extensive Output Fields	296
	Table 24: show task memory Output Fields	297
Chapter 7	RIP Monitoring	301
	Table 25: Summary of Key RIP Routing Output Fields	301

About the Documentation

- [Documentation and Release Notes on page xi](#)
- [Supported Platforms on page xi](#)
- [Using the Examples in This Manual on page xii](#)
- [Documentation Conventions on page xiii](#)
- [Documentation Feedback on page xv](#)
- [Requesting Technical Support on page xv](#)

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- [ACX Series](#)
- [T Series](#)
- [MX Series](#)
- [M Series](#)
- [J Series](#)
- [SRX Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the CLI User Guide.

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons





Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

J-Web GUI Conventions

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>

- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Overview

- [Introduction to RIP on page 3](#)
- [RIP Standards on page 9](#)

CHAPTER 1

Introduction to RIP

- [RIP Overview on page 3](#)
- [RIP Configuration Overview on page 8](#)

RIP Overview

RIP is an interior gateway protocol (IGP) that uses a distance-vector algorithm to determine the best route to a destination, using the hop count as the metric.

In a RIP network, each router's forwarding table is distributed among the nodes through the flooding of routing table information. Because topology changes are flooded throughout the network, every node maintains the same list of destinations. Packets are then routed to these destinations based on path-cost calculations done at each node in the network.



NOTE: In general, the term *RIP* refers to RIP version 1 and RIP version 2.

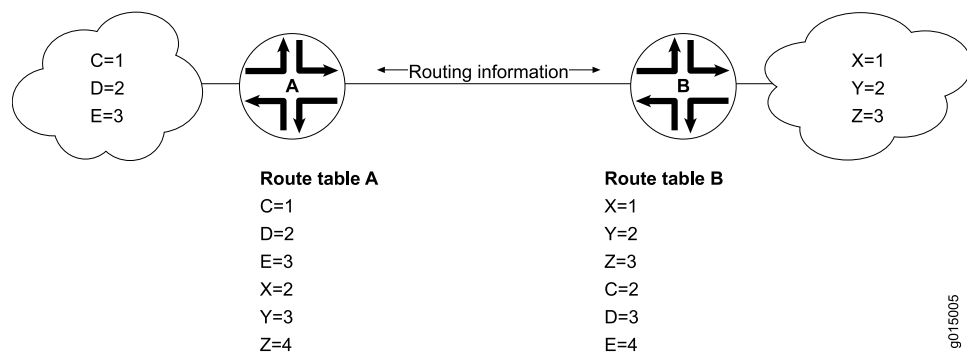
This topic contains the following sections:

- [Distance-Vector Routing Protocols on page 3](#)
- [RIP Protocol Overview on page 4](#)
- [RIP Packets on page 5](#)
- [Maximizing Hop Count on page 6](#)
- [Split Horizon and Poison Reverse Efficiency Techniques on page 6](#)
- [Limitations of Unidirectional Connectivity on page 7](#)

Distance-Vector Routing Protocols

Distance-vector routing protocols transmit routing information that includes a distance vector, typically expressed as the number of hops to the destination. This information is flooded out all protocol-enabled interfaces at regular intervals (every 30 seconds in the case of RIP) to create a network map that is stored in each node's local topology database. [Figure 1 on page 4](#) shows how distance-vector routing works.

Figure 1: Distance-Vector Protocol



In [Figure 1 on page 4](#), Routers A and B have RIP enabled on adjacent interfaces. Router A has known RIP neighbors Routers C, D, and E, which are 1, 2, and 3 hops away, respectively. Router B has known RIP neighbors Routers X, Y, and Z, which are 1, 2, and 3 hops away, respectively. Every 30 seconds, each router floods its entire routing table information out all RIP-enabled interfaces. In this case, flooding exchanges routing table information across the RIP link.

When Router A receives routing information from Router B, it adds 1 to the hop count to determine the new hop count. For example, Router X has a hop count of 1, but when Router A imports the route to X, the new hop count is 2. The imported route also includes information about where the route was learned, so that the original route is imported as a route to Router X through Router B with a hop count of 2.

When multiple routes to the same host are received, RIP uses the distance-vector algorithm to determine which path to import into the forwarding table. The route with the smallest hop count is imported. If there are multiple routes with the same hop count, all are imported into the forwarding table, and traffic is sent along the paths in round-robin fashion.

RIP Protocol Overview

The RIP IGP uses the Bellman-Ford, or *distance-vector*, algorithm to determine the best route to a destination. RIP uses the hop count as the metric. RIP enables hosts and routers to exchange information for computing routes through an IP-based network. RIP is intended to be used as an IGP in reasonably homogeneous networks of moderate size.

The Junos[®] operating system (Junos OS) supports RIP versions 1 and 2.



NOTE: RIP is not supported for multipoint interfaces.

RIP version 1 packets contain the minimal information necessary to route packets through a network. However, this version of RIP does not support authentication or subnetting.

RIP uses User Datagram Protocol (UDP) port 520.

RIP has the following architectural limitations:

- The longest network path cannot exceed 15 hops (assuming that each network, or hop, has a cost of 1).
- RIP depends on counting to infinity to resolve certain unusual situations—When the network consists of several hundred routers, and when a routing loop has formed, the amount of time and network bandwidth required to resolve a next hop might be great.
- RIP uses only a fixed metric to select a route. Other IGPs use additional parameters, such as measured delay, reliability, and load.

RIP Packets

RIP packets contain the following fields:

- Command—Indicates whether the packet is a request or response message. Request messages seek information for the router's routing table. Response messages are sent periodically and also when a request message is received. Periodic response messages are called *update messages*. Update messages contain the command and version fields and 25 destinations (by default), each of which includes the destination IP address and the metric to reach that destination.



NOTE: Beginning with Junos OS Release 11.1, three additional command field types are available to support RIP demand circuits. When you configure an interface for RIP demand circuits, the command field indicates whether the packet is an update request, update response, or update acknowledge message. Neighbor interfaces send updates on demand, not periodically. These command field types are only valid on interfaces configured for RIP demand circuits. For more detailed information, see [“RIP Demand Circuits Overview” on page 77](#).

- Version number—Version of RIP that the originating router is running.
- Address family identifier—Address family used by the originating router. The family is always IP.
- Address—IP address included in the packet.
- Metric—Value of the metric advertised for the address.
- Mask—Mask associated with the IP address (RIP version 2 only).
- Next hop—IP address of the next-hop router (RIP version 2 only).

Routing information is exchanged in a RIP network by RIP request and RIP response packets. A router that has just booted can broadcast a RIP request on all RIP-enabled interfaces. Any routers running RIP on those links receive the request and respond by sending a RIP response packet immediately to the router. The response packet contains the routing table information required to build the local copy of the network topology map.

In the absence of RIP request packets, all RIP routers broadcast a RIP response packet every 30 seconds on all RIP-enabled interfaces. The RIP broadcast is the primary way in which topology information is flooded throughout the network.

Once a router learns about a particular destination through RIP, it starts a timer. Every time it receives a new response packet with information about the destination, the router resets the timer to zero. However, if the router receives no updates about a particular destination for 180 seconds, it removes the destination from its RIP routing table.

In addition to the regular transmission of RIP packets every 30 seconds, if a router detects a new neighbor or detects that an interface is unavailable, it generates a triggered update. The new routing information is immediately broadcast out all RIP-enabled interfaces, and the change is reflected in all subsequent RIP response packets.

Maximizing Hop Count

The successful routing of traffic across a RIP network requires that every node in the network maintain the same view of the topology. Topology information is broadcast between RIP neighbors every 30 seconds. If Router A is many hops away from a new host, Router B, the route to B might take significant time to propagate through the network and be imported into Router A's routing table. If the two routers are 5 hops away from each other, Router A cannot import the route to Router B until 2.5 minutes after Router B is online (30 seconds per hop). For large numbers of hops, the delay becomes prohibitive. To help prevent this delay from growing arbitrarily large, RIP enforces a maximum hop count of 15 hops. Any prefix that is more than 15 hops away is treated as unreachable and assigned a hop count equal to infinity. This maximum hop count is called the *network diameter*.

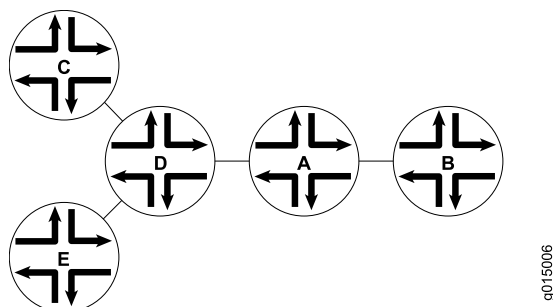
Split Horizon and Poison Reverse Efficiency Techniques

Because RIP functions by periodically flooding the entire routing table out to the network, it generates a lot of traffic. The split horizon and poison reverse techniques can help reduce the amount of network traffic originated by RIP hosts and make the transmission of routing information more efficient.

If a router receives a set of route advertisements on a particular interface, RIP determines that those advertisements do not need to be retransmitted out the same interface. This technique, known as *split horizon*, helps limit the amount of RIP routing traffic by eliminating information that other neighbors on that interface have already learned.

[Figure 2 on page 6](#) shows an example of the split horizon technique.

Figure 2: Split Horizon Example



Similarly, the poison reverse technique helps to optimize the transmission of routing information and improve the time to reach network convergence. If Router A learns about unreachable routes through one of its interfaces, it advertises those routes as unreachable (hop count of 16) out the same interface. [Figure 3 on page 7](#) shows an example of the poison reverse technique.



Sectional Connectivity

Figure 4: Limitations of Unidirectional Connectivity



In [Figure 4 on page 7](#), Routers A and D flood their routing table information to Router B. Because the path to Router E has the fewest hops when routed through Router A, that route is imported into Router B's forwarding table. However, suppose that Router A can transmit traffic but is not receiving traffic from Router B because of an unavailable link or invalid routing policy. If the only route to Router E is through Router A, any traffic destined for Router A is lost, because bidirectional connectivity was never established.

OSPF establishes bidirectional connectivity with a three-way handshake.

**Related
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [RIP Configuration Overview on page 8](#)
- [Example: Configuring RIP on page 13](#)

RIP Configuration Overview

To achieve basic connectivity between all RIP hosts in a RIP network, you enable RIP on every interface that is expected to transmit and receive RIP traffic, as described in the steps that follow.

To configure a RIP network:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Define RIP groups, which are logical groupings of interfaces, and add interfaces to the groups. Then, configure a routing policy to export directly connected routes and routes learned through RIP routing exchanges. See [“Example: Configuring a Basic RIP Network” on page 14](#).
3. (Optional) Configure metrics to control traffic through the RIP network. See [“Example: Controlling Traffic in a RIP Network with an Incoming Metric” on page 53](#) and [“Example: Controlling Traffic in a RIP Network with an Outgoing Metric” on page 55](#).
4. (Optional) Configure authentication to ensure that only trusted routers participate in the autonomous system's routing. See [“Enabling Authentication with Plain-Text Passwords \(CLI Procedure\)” on page 25](#) and [“Enabling Authentication with MD5 Authentication \(CLI Procedure\)” on page 25](#).

**Related
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [RIP Overview on page 3](#)
- [Verifying a RIP Configuration on page 94](#)

CHAPTER 2

RIP Standards

- [Supported RIP and RIPng Standards on page 9](#)

Supported RIP and RIPng Standards

Junos OS substantially supports the following RFCs, which define standards for RIP (for IP version 4 [IPv4]) and RIP next generation (RIPng, for IP version 6 [IPv6]).

Junos OS supports authentication for all RIP protocol exchanges (MD5 or simple authentication).

- RFC 1058, *Routing Information Protocol*
- RFC 2080, *RIPng for IPv6*
- RFC 2082, *RIP-2 MD5 Authentication*

Multiple keys using distinct key IDs are not supported.

- RFC 2453, *RIP Version 2*

The following RFC does not define a standard, but provides information about RIPng. The IETF classifies it as “Informational.”

- RFC 2081, *RIPng Protocol Applicability Statement*

Related Documentation

- [Supported IPv4, TCP, and UDP Standards](#)
- [Supported IPv6 Standards](#)
- [Accessing Standards Documents on the Internet](#)

PART 2

Configuration

- [Concepts and Examples on page 13](#)
- [Configuration Statements on page 99](#)

CHAPTER 3

Concepts and Examples

- [Example: Configuring RIP on page 13](#)
- [Example: Configuring Authentication for RIP Routes on page 20](#)
- [Example: Configuring BFD for RIP on page 26](#)
- [Example: Configuring BFD Authentication for RIP on page 32](#)
- [Example: Configuring Point-to-Multipoint RIP Networks on page 40](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 47](#)
- [Examples: Controlling Traffic with Metrics in a RIP Network on page 52](#)
- [Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 60](#)
- [Example: Redistributing Routes Among RIP Instances on page 64](#)
- [Example: Configuring RIP Timers on page 70](#)
- [Example: Configuring RIP Demand Circuits on page 77](#)
- [Example: Configuring Point-to-Multipoint RIP Networks on page 82](#)
- [Example: Tracing RIP Protocol Traffic on page 89](#)
- [Verifying a RIP Configuration on page 94](#)

Example: Configuring RIP

- [Understanding Basic RIP Routing on page 13](#)
- [Example: Configuring a Basic RIP Network on page 14](#)

Understanding Basic RIP Routing

RIP is an interior gateway protocol (IGP) that routes packets within a single autonomous system (AS). By default, RIP does not advertise the subnets that are directly connected through the device's interfaces. For traffic to pass through a RIP network, you must create a routing policy to export these routes. Advertising only the direct routes propagates the routes to the immediately adjacent RIP-enabled router only. To propagate all routes through the entire RIP network, you must configure the routing policy to export the routes learned through RIP.

Example: Configuring a Basic RIP Network

This example shows how to configure a basic RIP network.

- [Requirements on page 14](#)
- [Overview on page 14](#)
- [Configuration on page 14](#)
- [Verification on page 17](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

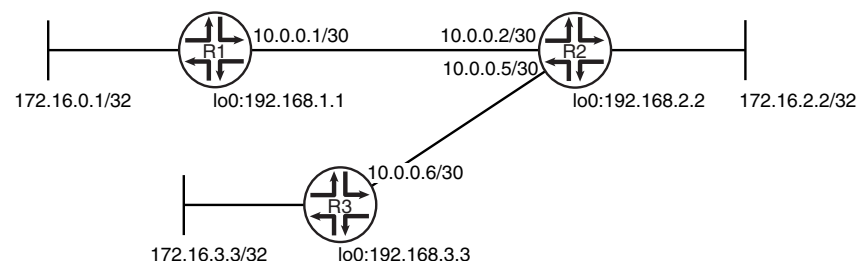
In this example, you configure a basic RIP network, create a RIP group called **rip-group**, and add the directly connected interfaces to the RIP group. Then you configure a routing policy to advertise direct routes using policy statement **advertise-routes-through-rip**.

By default, Junos OS does not advertise RIP routes, not even routes that are learned through RIP. To advertise RIP routes, you must configure and apply an export routing policy that advertises RIP-learned and direct routes.

In Junos OS, you do not need to configure the RIP version. RIP version 2 is used by default.

To use RIP on the device, you must configure RIP on all of the RIP interfaces within the network. [Figure 5 on page 14](#) shows the topology used in this example.

Figure 5: Sample RIP Network Topology



“CLI Quick Configuration” on page 14 shows the configuration for all of the devices in [Figure 5 on page 14](#). The section “Step-by-Step Procedure” on page 15 describes the steps on Device R1.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
```

```

set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To configure a basic RIP network:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

```

```

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]

```

```
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking the Routing Table on page 17](#)
- [Looking at the Routes That Device R1 Is Advertising to Device R2 on page 17](#)
- [Looking at the Routes That Device R1 Is Receiving from Device R2 on page 18](#)
- [Verifying the RIP-Enabled Interfaces on page 18](#)
- [Verifying the Exchange of RIP Messages on page 18](#)
- [Verifying Reachability of All Hosts in the RIP Network on page 19](#)

Checking the Routing Table

Purpose Verify that the routing table is populated with the expected routes..

Action From operational mode, enter the **show route protocol rip** command.

```
user@R1> show route protocol rip
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[RIP/100] 00:59:15, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32   *[RIP/100] 02:52:48, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32   *[RIP/100] 00:45:05, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32  *[RIP/100] 02:52:48, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32  *[RIP/100] 00:45:05, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
224.0.0.9/32    *[RIP/100] 00:45:09, metric 1
                 MultiRecv
```

Meaning The output shows that the routes have been learned from Device R2 and Device R3.

If you were to delete the **from protocol rip** condition in the routing policy on Device R2, the remote routes from Device R3 would not be learned on Device R1.

Looking at the Routes That Device R1 Is Advertising to Device R2

Purpose Verify that Device R1 is sending the expected routes.

Action From operational mode, enter the **show route advertising-protocol rip** command.

```
user@R1> show route advertising-protocol rip 10.0.0.1
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.1/32   *[Direct/0] 05:18:26
                 > via lo0.1
192.168.1.1/32  *[Direct/0] 05:18:25
                 > via lo0.1
```

Meaning Device R1 is sending routes to its directly connected networks.

Looking at the Routes That Device R1 Is Receiving from Device R2

Purpose Verify that Device R1 is receiving the expected routes.

Action From operational mode, enter the **show route receive-protocol rip** command.

```
user@R1> show route receive-protocol rip 10.0.0.2
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30          *[RIP/100] 02:31:22, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32       *[RIP/100] 04:24:55, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32       *[RIP/100] 02:17:12, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32      *[RIP/100] 04:24:55, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32      *[RIP/100] 02:17:12, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
```

Meaning Device R1 is receiving from Device R2 all of Device R2's directly connected networks. Device R1 is also receiving from Device R2 all of Device R3's directly connected networks, which Device R2 learned from Device R3 through RIP.

Verifying the RIP-Enabled Interfaces

Purpose Verify that all RIP-enabled Interfaces are available and active.

Action From operational mode, enter the **show rip neighbor** command.

```
user@R1> show rip neighbor
```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-1/2/0.1	Up	10.0.0.1	224.0.0.9	mcast	both	1

Meaning The output shows that the RIP-enabled interface on Device R1 is operational.

In general for this command, the output shows a list of the RIP neighbors that are configured on the device. Verify the following information:

- Each configured interface is present. Interfaces are listed in alphabetical order.
- Each configured interface is up. The state of the interface is listed in the **Local State** column. A state of **Up** indicates that the link is passing RIP traffic. A state of **Dn** indicates that the link is not passing RIP traffic. In a point-to-point link, this state generally means that either the end point is not configured for RIP or the link is unavailable.

Verifying the Exchange of RIP Messages

Purpose Verify that RIP messages are being sent and received on all RIP-enabled interfaces.

Action From operational mode, enter the **show rip statistics** command.

```
user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped
              5              0              0              0

fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval
30s
Counter              Total    Last 5 min  Last minute
-----
Updates Sent          2669         10         2
Triggered Updates Sent      2         0         0
Responses Sent          0         0         0
Bad Messages           0         0         0
RIPv1 Updates Received     0         0         0
RIPv1 Bad Route Entries    0         0         0
RIPv1 Updates Ignored      0         0         0
RIPv2 Updates Received    2675        11         2
RIPv2 Bad Route Entries    0         0         0
RIPv2 Updates Ignored      0         0         0
Authentication Failures     0         0         0
RIP Requests Received      0         0         0
RIP Requests Ignored       0         0         0
none                      0         0         0
```

Meaning The output shows the number of RIP routes learned. It also shows the number of RIP updates sent and received on the RIP-enabled interfaces. Verify the following information:

- The number of RIP routes learned matches the number of expected routes learned. Subnets learned by direct connectivity through an outgoing interface are not listed as RIP routes.
- RIP updates are being sent on each RIP-enabled interface. If no updates are being sent, the routing policy might not be configured to export routes.
- RIP updates are being received on each RIP-enabled interface. If no updates are being received, the routing policy might not be configured to export routes on the host connected to that subnet. The lack of updates might also indicate an authentication error.

Verifying Reachability of All Hosts in the RIP Network

Purpose Use the **traceroute** command on each loopback address in the network to verify that all hosts in the RIP network are reachable from each Juniper Networks device.

Action From operational mode, enter the **traceroute** command.

```
user@R1> traceroute 192.168.3.3
traceroute to 192.168.3.3 (192.168.3.3), 30 hops max, 40 byte packets
 1  10.0.0.2 (10.0.0.2)  1.094 ms  1.028 ms  0.957 ms
 2  192.168.3.3 (192.168.3.3)  1.344 ms  2.245 ms  2.125 ms
```

Meaning Each numbered row in the output indicates a routing hop in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop for each traceroute packet.

To ensure that the RIP network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.
- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is probably unreachable. It might also indicate that the incoming or outgoing metric on one or more hosts has been set unexpectedly.

**Related
Documentation**

- [Example: Configuring Point-to-Multipoint RIP Networks on page 40](#)

Example: Configuring Authentication for RIP Routes

- [Understanding RIP Authentication on page 20](#)
- [Example: Configuring Route Authentication for RIP on page 20](#)
- [Enabling Authentication with Plain-Text Passwords \(CLI Procedure\) on page 25](#)
- [Enabling Authentication with MD5 Authentication \(CLI Procedure\) on page 25](#)

Understanding RIP Authentication

RIPv2 provides authentication support so that RIP links can require authentication keys (passwords) before they become active. Authentication provides an additional layer of security on the network beyond the other security features. By default, this authentication is disabled.

Authentication keys can be specified in either plain-text or MD5 form. Authentication requires all routers within the RIP network or subnetwork to have the same authentication type and key (password) configured.

This type of authentication is not supported on RIPv1 networks.

Example: Configuring Route Authentication for RIP

This example shows how to configure authentication for a RIP network.

- [Requirements on page 20](#)
- [Overview on page 20](#)
- [Configuration on page 21](#)
- [Verification on page 24](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

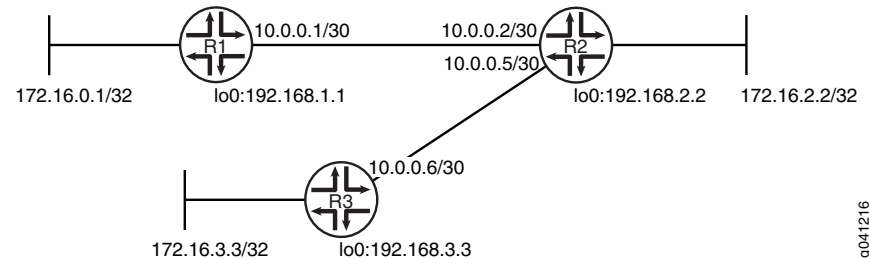
You can configure the router to authenticate RIP route queries. By default, authentication is disabled. You can use one of the following authentication methods:

- Simple authentication—Uses a text password that is included in the transmitted packet. The receiving router uses an authentication key (password) to verify the packet.
- MD5 authentication—Creates an encoded checksum that is included in the transmitted packet. The receiving router uses an authentication key (password) to verify the packet's MD5 checksum.

This example shows MD5 authentication.

Figure 6 on page 21 shows the topology used in this example.

Figure 6: RIP Authentication Network Topology



"CLI Quick Configuration" on page 21 shows the configuration for all of the devices in Figure 6 on page 21. The section "Step-by-Step Procedure" on page 22 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$ONLRBhreK87dsM8i.5FAtM8XxNb"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2

```

```
set protocols rip group rip-group neighbor fe-1/2/1.5
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$Lf1Xds2gJDHmoJCu1hKvoJGUjq"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$G.UkP5T39tOz3K87V4oz36/Cu"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure RIP authentication:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

```
user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
```

```
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

5. Require MD5 authentication for RIP route queries received on an interface.

The passwords must match on neighboring RIP routers. If the password does not match, the packet is rejected. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.

Do not enter the password as shown here. The password shown here is the encrypted password that is displayed in the configuration after the actual password is already configured.

```
[edit protocols rip]
user@R1# set authentication-type md5
user@R1# set authentication-key "$9$ONLRBhreK87dsM8i.5FAtM8XxNb"
```

6. Configure tracing operations to track authentication.

```
[edit protocols rip traceoptions]
user@R1# set file rip-authentication-messages
user@R1# set flag auth
user@R1# set flag packets
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  traceoptions {
    file rip-authentication-messages;
    flag auth;
```

```

        flag packets;
    }
    authentication-type md5;
    authentication-key "$9$ONLRBhreK87dsM8i.5FAtM8XxNb"; ## SECRET-DATA
    group rip-group {
        export advertise-routes-through-rip;
        neighbor fe-1/2/0.1;
    }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking for Authentication Failures on page 24](#)
- [Verifying That MD5 Authentication Is Enabled in RIP Update Packets on page 25](#)

Checking for Authentication Failures

Purpose Verify that there are no authentication failures.

Action From operational mode, enter the **show rip statistics** command.

```

user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped
              5              0              0              0

fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval
30s
Counter              Total    Last 5 min  Last minute
-----
Updates Sent          2669         10           2
Triggered Updates Sent      2           0           0
Responses Sent          0           0           0
Bad Messages           0           0           0
RIPv1 Updates Received     0           0           0
RIPv1 Bad Route Entries    0           0           0
RIPv1 Updates Ignored      0           0           0
RIPv2 Updates Received    2675         11           2
RIPv2 Bad Route Entries    0           0           0
RIPv2 Updates Ignored      0           0           0
Authentication Failures     0           0           0
RIP Requests Received      0           0           0
RIP Requests Ignored       0           0           0
none                     0           0           0

```

Meaning The output shows that there are no authentication failures.

Verifying That MD5 Authentication Is Enabled in RIP Update Packets

Purpose Use tracing operations to verify that MD5 authentication is enabled in RIP updates.

Action From operational mode, enter the **show log** command.

```
user@R1> show log rip-authentication-messages | match md5
Feb 15 15:45:13.969462      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:45:43.229867      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:46:13.174410      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:46:42.716566      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:47:11.425076      sending msg 0xb9a8c04, 3 rtes (needs MD5)
...
```

Meaning The **(needs MD5)** output shows that all route updates require MD5 authentication.

Enabling Authentication with Plain-Text Passwords (CLI Procedure)

To configure authentication that requires a plain-text password to be included in the transmitted packet, enable simple authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in [Table 3 on page 25](#).
3. If you are finished configuring the router, commit the configuration.

Table 3: Configuring Simple RIP Authentication

Task	CLI Configuration Editor
Navigate to Rip level in the configuration hierarchy.	From the [edit] hierarchy level, enter edit protocols rip
Set the authentication type to simple .	Set the authentication type to simple : set authentication-type simple
Set the authentication key to a simple-text password. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the authentication key to a simple-text password: set authentication-key <i>password</i>

Enabling Authentication with MD5 Authentication (CLI Procedure)

To configure authentication that requires an MD5 password to be included in the transmitted packet, enable MD5 authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in [Table 4 on page 26](#).

3. If you are finished configuring the router, commit the configuration.

Table 4: Configuring MD5 RIP Authentication

Task	CLI Configuration Editor
Navigate to Rip level in the configuration hierarchy.	From the [edit] hierarchy level, enter edit protocols rip
Set the authentication type to MD5 .	Set the authentication type to md5 : set authentication-type md5
Set the MD5 authentication key (password). The key can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the MD5 authentication key: set authentication-key <i>password</i>

Related Documentation

- [Example: Configuring RIP on page 13](#)

Example: Configuring BFD for RIP

- [Understanding BFD for RIP on page 26](#)
- [Example: Configuring BFD for RIP on page 27](#)

Understanding BFD for RIP

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. BFD failure detection times are shorter than RIP detection times, providing faster reaction times to various kinds of failures in the network. Instead of waiting for the routing protocol neighbor timeout, BFD provides rapid detection of link failures. BFD timers are adaptive and can be adjusted to be more or less aggressive. For example, a timer can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the one configured.

BFD enables quick failover between a primary and a secondary routed path. The protocol tests the operational status of the interface multiple times per second. BFD provides for configuration timers and thresholds for failure detection. For example, if the minimum interval is set for 50 milliseconds and the threshold uses the default value of three missed messages, a failure is detected on an interface within 200 milliseconds of the failure.

Intervening devices (for example, an Ethernet LAN switch) hide link-layer failures from routing protocol peers, such as when two routers are connected by way of a LAN switch, where the local interface status remains up even when a physical fault happens on the remote link. Link-layer failure detection times vary, depending on the physical media and the Layer 2 encapsulation. BFD can provide fast failure detection times for all media types, encapsulations, topologies, and routing protocols.

To enable BFD for RIP, both sides of the connection must receive an update message from the peer. By default, RIP does not export any routes. Therefore, you must enable update messages to be sent by configuring an export policy for routes before a BFD session is triggered.

Example: Configuring BFD for RIP

This example shows how to configure Bidirectional Forwarding Detection (BFD) for a RIP network.

- [Requirements on page 27](#)
- [Overview on page 27](#)
- [Configuration on page 29](#)
- [Verification on page 31](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

To enable failure detection, include the **bfd-liveness-detection** statement:

```
bfd-liveness-detection {  
  detection-time {  
    threshold milliseconds;  
  }  
  minimum-interval milliseconds;  
  minimum-receive-interval milliseconds;  
  multiplier number;  
  no-adaptation;  
  transmit-interval {  
    threshold milliseconds;  
    minimum-interval milliseconds;  
  }  
  version (1 | automatic);  
}
```

Optionally, you can specify the threshold for the adaptation of the detection time by including the **threshold** statement. When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a system log message are sent.

To specify the minimum transmit and receive interval for failure detection, include the **minimum-interval** statement. This value represents the minimum interval at which the local routing device transmits hello packets as well as the minimum interval at which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds. This examples sets a minimum interval of 600 milliseconds.



NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD of less than 100 ms for Routing Engine-based sessions and 10 ms for distributed BFD sessions can cause undesired BFD flapping.

Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

You can optionally specify the minimum transmit and receive intervals separately.

To specify only the minimum receive interval for failure detection, include the **minimum-receive-interval** statement. This value represents the minimum interval at which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify only the minimum transmit interval for failure detection, include the **transmit-interval minimum-interval** statement. This value represents the minimum interval at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify the number of hello packets not received by a neighbor that causes the originating interface to be declared down, include the **multiplier** statement. The default is 3, and you can configure a value in the range from 1 through 255.

To specify the threshold for detecting the adaptation of the transmit interval, include the **transmit-interval threshold** statement. The threshold value must be greater than the transmit interval.

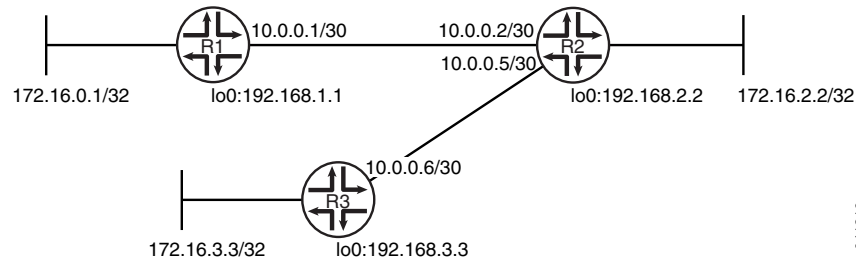
To specify the BFD version used for detection, include the **version** statement. The default is to have the version detected automatically.

You can trace BFD operations by including the **traceoptions** statement at the **[edit protocols bfd]** hierarchy level.

In Junos OS Release 9.0 and later, you can configure BFD sessions not to adapt to changing network conditions. To disable BFD adaptation, include the **no-adaptation** statement. We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

Figure 7 on page 29 shows the topology used in this example.

Figure 7: RIP BFD Network Topology



“CLI Quick Configuration” on page 29 shows the configuration for all of the devices in Figure 7 on page 29. The section “Step-by-Step Procedure” on page 30 describes the steps on Device R1.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set protocols bfd traceoptions file bfd-trace set protocols bfd traceoptions flag all set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 set protocols rip group rip-group neighbor fe-1/2/1.5 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R3	<pre> set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30 set protocols rip group rip-group export advertise-routes-through-rip </pre>

```
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

**Step-by-Step
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure a BFD for a RIP network:

1. Configure the network interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

5. Enable BFD.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection minimum-interval 600
```

6. Configure tracing operations to track BFD messages.

```
[edit protocols bfd traceoptions]
user@R1# set file bfd-trace
user@R1# set flag all
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
```

```

unit 1 {
    family inet {
        address 10.0.0.1/30;
    }
}

user@R1# show protocols
bfd {
    traceoptions {
        file bfd-trace;
        flag all;
    }
}
rip {
    group rip-group {
        export advertise-routes-through-rip;
        bfd-liveness-detection {
            minimum-interval 600;
        }
        neighbor fe-1/2/0.1;
    }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the BFD Sessions Are Up on page 31](#)
- [Checking the BFD Trace File on page 32](#)

Verifying That the BFD Sessions Are Up

Purpose Make sure that the BFD sessions are operating.

Action From operational mode, enter the **show bfd session** command.

```

user@R1> show bfd session

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

1 sessions, 1 clients
Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

Meaning The output shows that there are no authentication failures.

Checking the BFD Trace File

Purpose Use tracing operations to verify that BFD packets are being exchanged.

Action From operational mode, enter the **show log** command.

```
user@R1> show log bfd-trace
Feb 16 10:26:32 PPM Trace: BFD periodic xmit to 10.0.0.2 (IFL 124, rtbl 53,
single-hop port)
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 86:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 61: (hex) 42 46 44 20 70 61 63 6b 65 74 20 66 72
6f 6d 20 31 30 2e
Feb 16 10:26:32 PPM Trace: BFD packet from 10.0.0.1 (IFL 73, rtbl 56, ttl 255)
absorbed
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 60:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
78 6d 69 74 20 6f
...
```

Meaning The output shows the normal functioning of BFD.

- Related Documentation**
- [Example: Configuring RIP on page 13](#)
 - [Example: Configuring Authentication for RIP Routes on page 20](#)
 - [Example: Configuring Point-to-Multipoint RIP Networks on page 40](#)

Example: Configuring BFD Authentication for RIP

- [Understanding BFD Authentication for RIP on page 32](#)
- [Example: Configuring BFD Authentication for RIP on page 34](#)

Understanding BFD Authentication for RIP

BFD enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when running BFD over Network Layer protocols, the risk of service attacks can be significant. We strongly recommend using authentication if you are running BFD over multiple hops or through insecure tunnels. Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over RIP. BFD authentication is only supported in the domestic image and is not available in the export image.

You authenticate BFD sessions by specifying an authentication algorithm and keychain, and then associating that configuration information with a security authentication keychain using the keychain name.

The following sections describe the supported authentication algorithms, security keychains, and the level of authentication that can be configured:

- [BFD Authentication Algorithms on page 33](#)
- [Security Authentication Keychains on page 33](#)
- [Strict Versus Loose Authentication on page 34](#)

BFD Authentication Algorithms

Junos OS supports the following algorithms for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.
- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method might take additional time to authenticate the session.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method might take additional time to authenticate the session.



NOTE: Nonstop active routing is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

Security Authentication Keychains

The security authentication keychain defines the authentication attributes used for authentication key updates. When the security authentication keychain is configured and

associated with a protocol through the keychain name, authentication key updates can occur without interrupting routing and signaling protocols.

The authentication keychain contains one or more keychains. Each keychain contains one or more keys. Each key holds the secret data and the time at which the key becomes valid. The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

BFD allows multiple clients per session, and each client can have its own keychain and algorithm defined. To avoid confusion, we recommend specifying only one security authentication keychain.

Strict Versus Loose Authentication

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session. This feature is intended for transitional periods only.

Example: Configuring BFD Authentication for RIP

This example shows how to configure Bidirectional Forwarding Detection (BFD) authentication for a RIP network.

- [Requirements on page 34](#)
- [Overview on page 34](#)
- [Configuration on page 35](#)
- [Verification on page 39](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

The devices must be running Junos OS Release 9.6 or later.

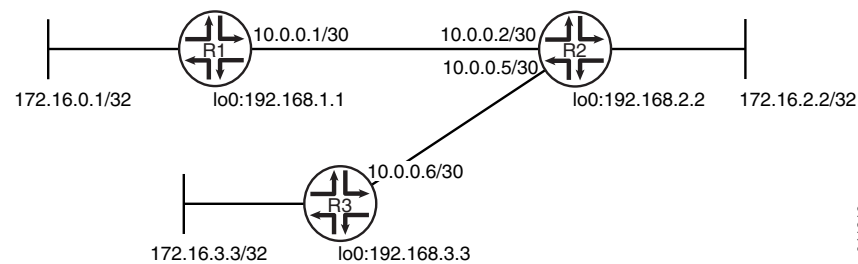
Overview

Only three steps are needed to configure authentication on a BFD session:

1. Specify the BFD authentication algorithm for the RIP protocol.
2. Associate the authentication keychain with the RIP protocol.
3. Configure the related security authentication keychain.

[Figure 8 on page 35](#) shows the topology used in this example.

Figure 8: RIP BFD Authentication Network Topology



"CLI Quick Configuration" on page 35 shows the configuration for all of the devices in Figure 8 on page 35. The section "Step-by-Step Procedure" on page 36 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip
set protocols rip group rip-group bfd-liveness-detection authentication algorithm
    keyed-md5
set protocols rip group rip-group bfd-liveness-detection authentication loose-check
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
    direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
    rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set security authentication-key-chains key-chain bfd-rip key 53 secret
    "$9$d1V2aZGi.fzDiORSeXxDikqmT"
set security authentication-key-chains key-chain bfd-rip key 53 start-time
    "2012-2-16.12:00:00 -0800"

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip
set protocols rip group rip-group bfd-liveness-detection authentication algorithm
    keyed-md5
set protocols rip group rip-group bfd-liveness-detection authentication loose-check
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
    direct

```

```
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set security authentication-key-chains key-chain bfd-rip key 53 secret
  "$9$d1V2aZGi.fzDiORSeXxDikqmT"
set security authentication-key-chains key-chain bfd-rip key 53 start-time
  "2012-2-16.12:00:00 -0800"
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip
set protocols rip group rip-group bfd-liveness-detection authentication algorithm
  keyed-md5
set protocols rip group rip-group bfd-liveness-detection authentication loose-check
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set security authentication-key-chains key-chain bfd-rip key 53 secret
  "$9$d1V2aZGi.fzDiORSeXxDikqmT"
set security authentication-key-chains key-chain bfd-rip key 53 start-time
  "2012-2-16.12:00:00 -0800"
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To configure a BFD authentication:

1. Configure the network interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

5. Enable BFD.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection minimum-interval 600
```

6. Specify the algorithm (**keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**, or **simple-password**) to use.



NOTE: Nonstop active routing is not supported with **meticulous-keyed-md5** and **meticulous-keyed-sha-1** authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication algorithm keyed-md5
```

7. Specify the keychain to be used to associate BFD sessions on RIP with the unique security authentication keychain attributes.

The keychain you specify must match a keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication key-chain bfd-rip
```

8. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication loose-check
```

9. Specify the unique security authentication information for BFD sessions:

- The matching keychain name as specified in Step 7.
- At least one key, a unique integer between 0 and 63. Creating multiple keys allows multiple clients to use the BFD session.
- The secret data used to allow access to the session.
- The time at which the authentication key becomes active, in the format *yyyy-mm-dd.hh:mm:ss*.

```
[edit security authentication-key-chains key-chain bfd-rip]
user@R1# set key 53 secret "$9$d1V2aZGi.fzDiORSeXxDikqmT"
user@R1# set key 53 start-time "2012-2-16.12:00:00 -0800"
```

10. Configure tracing operations to track BFD authentication.

```
[edit protocols bfd traceoptions]
user@R1# set file bfd-trace
user@R1# set flag all
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show security** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}

user@R1# show protocols
bfd {
  traceoptions {
    file bfd-trace;
    flag all;
  }
}
rip {
  group rip-group {
    export advertise-routes-through-rip;
    bfd-liveness-detection {
      minimum-interval 600;
    }
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

user@R1# show security
authentication-key-chains {
  key-chain bfd-rip {
    key 53 {
      secret "$9$d1V2aZGi.fzDiORSeXxDikqmT"; ## SECRET-DATA
      start-time "2012-2-16.12:00:00 -0800";
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the BFD Sessions Are Authenticated on page 39](#)
- [Viewing Extensive Information About the BFD Authentication on page 39](#)
- [Checking the BFD Trace File on page 40](#)

Verifying That the BFD Sessions Are Authenticated

Purpose Make sure that the BFD sessions are authenticated.

Action From operational mode, enter the **show bfd session detail** command.

```
user@R1> show bfd session detail
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

Client RIP, TX interval 0.600, RX interval 0.600, **Authenticate**
 Session up time 01:39:34
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 6, routing table index 53

1 sessions, 1 clients
 Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

Meaning **Authenticate** is displayed to indicate that BFD authentication is configured.

Viewing Extensive Information About the BFD Authentication

Purpose View the keychain name, the authentication algorithm and mode for each client in the session, and the BFD authentication configuration status.

Action From operational mode, enter the **show bfd session extensive** command.

```
user@R1> show bfd session extensive
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

Client RIP, TX interval 0.600, RX interval 0.600, **Authenticate**
keychain bfd-rip, algo keyed-md5, mode loose
 Session up time 01:46:29
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 6, routing table index 53
 Min async interval 0.600, min slow interval 1.000
 Adaptive async TX interval 0.600, RX interval 0.600
 Local min TX interval 0.600, minimum RX interval 0.600, multiplier 3
 Remote min TX interval 0.600, min RX interval 0.600, multiplier 3
 Local discriminator 225, remote discriminator 226
 Echo mode disabled/inactive
Authentication enabled/active, keychain bfd-rip, algo keyed-md5, mode loose
 Session ID: 0x300501

1 sessions, 1 clients
 Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

Meaning The output shows the keychain name, the authentication algorithm and mode for the client in the session, and the BFD authentication configuration status.

Checking the BFD Trace File

Purpose Use tracing operations to verify that BFD packets are being exchanged.

Action From operational mode, enter the **show log** command.

```
user@R1> show log bfd-trace
Feb 16 10:26:32 PPM Trace: BFD periodic xmit to 10.0.0.2 (IFL 124, rtbl 53,
single-hop port)
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 86:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 61: (hex) 42 46 44 20 70 61 63 6b 65 74 20 66 72
6f 6d 20 31 30 2e
Feb 16 10:26:32 PPM Trace: BFD packet from 10.0.0.1 (IFL 73, rtbl 56, ttl 255)
absorbed
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 60:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
78 6d 69 74 20 6f
...
```

Meaning The output shows the normal functioning of BFD.

- Related Documentation**
- [Example: Configuring BFD for RIP on page 26](#)
 - [Example: Configuring Authentication for RIP Routes on page 20](#)
 - [Example: Configuring RIP on page 13](#)

Example: Configuring Point-to-Multipoint RIP Networks

- [Configuring Point-to-Multipoint RIP Networks Overview on page 40](#)
- [Example: Configuring Point-to-Multipoint RIP Networks on page 41](#)

Configuring Point-to-Multipoint RIP Networks Overview

A point-to-multipoint RIP network consists of a device having two or more peers on a single interface. All the devices forming a point-to-multipoint connection are placed in a single broadcast domain.

In a RIP network, a device can have a single peer or multiple peers for an interface. However, the demand circuit feature implementation in a RIP network requires the use of a single RIP peer. When you configure the following statements, a RIP network with demand circuits can also be configured to have multiple peers on an interface:

- Configuring the interface type to be a multipoint interface by using the [interface-type \(Protocols RIP\) p2mp](#) statement.

- Enabling dynamic peer discovery by using the [dynamic-peers](#) statement (SRX Series devices only).



NOTE: Before configuring the [dynamic-peers](#) statement, IPsec must be configured and IPsec tunnels must be set up by configuring IPsec parameters. Without IPsec configuration, the remote peers have to be explicitly configured at the RIP protocol level by using the [peer address](#) statement. See *Configuring Security Associations for IPsec on an ES PIC* for more details.

- Configuring peers by using the [peer address](#) statement.

```
[edit]
protocols {
  rip {
    group red {
      neighbor fe-0/1/3 {
        interface-type (Protocols RIP) p2mp;
        peer address; (or use dynamic-peers;)
      }
    }
  }
}
```

The [show rip statistics peer address](#) command can be used to display the RIP statistics at the peer level. The [clear rip statistics peer address](#) command can be used to clear the RIP statistics for a peer. Alternatively, you can use the [show rip statistics peer all](#) and [clear rip statistics peer all](#) command to display and clear RIP statistics for all peers.

Example: Configuring Point-to-Multipoint RIP Networks

This example shows how to configure a point-to-multipoint RIP network.

- [Requirements on page 41](#)
- [Overview on page 41](#)
- [Configuration on page 43](#)
- [Verification on page 45](#)

Requirements

This example uses the following hardware and software components:

- M Series routers, MX Series routers, T Series routers, or SRX Series devices
- Junos OS Release 12.1 or later

Overview

In a RIP network, a device can have a single peer or multiple peers for an interface. However, the demand circuit feature implementation in a RIP network requires the use of a single RIP peer.

When you include the following statements, the demand circuit implementation can have multiple peers for a given RIP neighbor.

- Configuring the interface type to be a multipoint interface by using the [interface-type \(Protocols RIP\) p2mp](#) statement.
- Enabling dynamic peer discovery by using the [dynamic-peers](#) statement (SRX Series devices only).



NOTE: To configure the [dynamic-peers](#) statement, IPsec tunnels must be set up by configuring IPsec parameters. See [Configuring Security Associations for IPsec on an ES PIC](#) for more details.

- Configuring peers by using the [peer address](#) statement.

```
[edit]
protocols {
  rip {
    group red {
      neighbor fe-0/1/3 {
        interface-type (Protocols RIP) p2mp;
        peer address; (or use dynamic-peers;)
      }
    }
  }
}
```

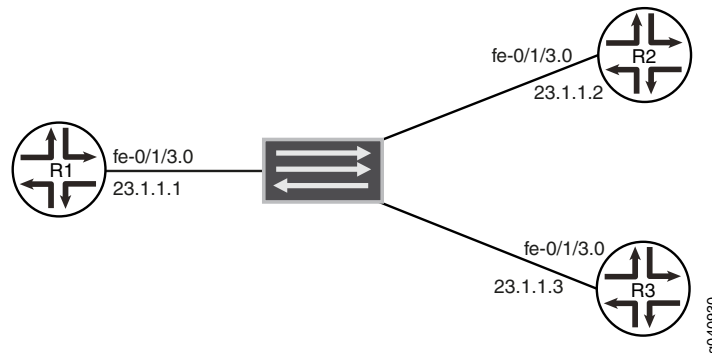
The [show rip statistics peer](#) command can be used to display the RIP statistics at the peer level.

Topology

In this example, Devices R1, R2, and R3 form a point-to-multipoint network. R1 is connected to R2 and to R3 as a point-to-multipoint connection through a switch that places all devices in the same broadcast domain. RIP demand circuits are configured on all three devices. The two peers to R1 are configured statically by using the [peer address](#) statement. The [dynamic-peers](#) statement is not used here.

[Figure 9 on page 43](#) shows the topology used in this example.

Figure 9: Configuring a Point-to-Multipoint RIP Network



Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-0/1/3 unit 0 family inet address 23.1.1/24
set policy-options policy-statement accept-rip-routes term from-direct from protocol direct
set policy-options policy-statement accept-rip-routes term from-direct then accept
set policy-options policy-statement accept-rip-routes term from-rip from protocol rip
set policy-options policy-statement accept-rip-routes term from-rip then accept
set protocols rip traceoptions file R1.log size 4m world-readable
set protocols rip traceoptions flag all detail
set protocols rip group red export accept-rip-routes
set protocols rip group red neighbor fe-0/1/3.0 interface-type p2mp
set protocols rip group red neighbor fe-0/1/3.0 peer 23.1.1.2
set protocols rip group red neighbor fe-0/1/3.0 peer 23.1.1.3
set protocols rip group red neighbor fe-0/1/3.0 demand-circuit
set protocols rip group red neighbor fe-0/1/3.0 max-retrans-time 10

```

Similarly, configure Devices R2 and R3, omitting the **peer address** configuration statement.

Configuring a Point-to-Multipoint RIP Network (with Demand Circuits)

Step-by-Step Procedure The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the point-to-multipoint feature across a RIP network:

1. Configure the device interface.

```

[edit interfaces fe-0/1/3 unit 0]
user@R1# set family inet address 23.1.1/24

```
2. Define a policy for exporting RIP routes from the routing table to the protocol for transmission through the network.

```

[edit policy-options policy-statement accept-rip-routes]
user@R1# set term from-direct from protocol direct

```

```

user@R1# set term from-direct then accept
user@R1# set term from-rip from protocol rip
user@R1# set term from-rip then accept

```

3. Configure RIP and a RIP group with the defined export policy and point-to-multipoint configuration statements.

```

[edit protocols rip]
user@R1# set traceoptions file R1.log size 4m world-readable
user@R1# set traceoptions flag all detail
user@R1# set group red export accept-rip-routes
user@R1# set group red neighbor fe-0/1/3.0 interface-type p2mp
user@R1# set group red neighbor fe-0/1/3.0 peer 23.1.1.2
user@R1# set group red neighbor fe-0/1/3.0 peer 23.1.1.3
user@R1# set group red neighbor fe-0/1/3.0 demand-circuit
user@R1# set group red neighbor fe-0/1/3.0 max-retrans-time 10

```

Similarly, configure Devices R2 and R3, omitting the **peer address** configuration statement.



NOTE: Configuring **max-retrans-time** is optional. In the absence of this configuration statement, the default retransmission time of 180 seconds is configured.

The configuration used in this example is for a RIP network with demand circuits. To configure RIP for networks without demand circuits, exclude the **demand-circuit** and **max-retrans-time** statements from the configuration and check the resulting output. For more information about configuring RIP demand circuits, see [“Example: Configuring RIP Demand Circuits” on page 79](#).

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, and **show protocols rip** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@R1# show interfaces
fe-0/1/3 {
  unit 0 {
    family inet {
      address 23.1.1.1/24;
    }
  }
}

user@R1# show protocols rip
traceoptions {
  file R1.log size 4m world-readable;
  flag all detail;
}
group red {

```

```
export accept-rip-routes;
neighbor fe-0/1/3.0 {
  interface-type p2mp;
  peer 23.1.1.2;
  peer 23.1.1.3;
  demand-circuit;
  max-retrans-time 10;
}
}

user@R1# show policy-options
policy-statement accept-rip-routes {
  term from-direct {
    from protocol direct;
    then accept;
  }
  term from-rip {
    from protocol rip;
    then accept;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying the Point-to-Multipoint RIP Network

Purpose Verify that the RIP network is functional with the point-to-multipoint feature configured.

Action From operational mode, run the **show rip neighbor** command.

```
user@R1> show rip neighbor
```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-0/1/3.0(DC)	Up	23.1.1.1	23.1.1.2	unicast	unicast	1
fe-0/1/3.0(DC)	Up	23.1.1.1	23.1.1.3	unicast	unicast	1

From operational mode, run the **show rip statistics peer address** command.

```
user@R1> show rip statistics peer 23.1.1.2
```

RIPv2 info: port 520; holddown 120s.

rts learned	rts held down	rqsts dropped	resps dropped
3	0	0	0

fe-0/1/3.0 Peer-Ip 23.1.1.2: 2 routes learned; 3 routes advertised; timeout 180s; update interval 0s

Counter	Total	Last 5 min	Last minute
Updates Sent	0	0	0
Triggered Updates Sent	3	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2	0	0
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	3	0	0

```
user@R1> show rip statistics peer 23.1.1.3
```

RIPv2 info: port 520; holddown 120s.

rts learned	rts held down	rqsts dropped	resps dropped
3	0	0	0

fe-0/1/3.0 Peer-Ip 23.1.1.3: 2 routes learned; 3 routes advertised; timeout 180s; update interval 0s

Counter	Total	Last 5 min	Last minute
Updates Sent	0	0	0
Triggered Updates Sent	3	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2	0	0
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0

none	3	0	0
------	---	---	---

Meaning The RIP network is up and running with the point-to-multipoint feature configured.

Related Documentation

- [Example: Configuring RIP on page 13](#)

Example: Applying Policies to RIP Routes Imported from Neighbors

- [Understanding RIP Import Policy on page 47](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 47](#)

Understanding RIP Import Policy

The default RIP import policy is to accept all received RIP routes that pass a sanity check. To filter routes being imported by the local routing device from its neighbors, include the **import** statement, and list the names of one or more policies to be evaluated. If you specify more than one policy, they are evaluated in order (first to last) and the first matching policy is applied to the route. If no match is found, the local routing device does not import any routes.

Example: Applying Policies to RIP Routes Imported from Neighbors

This example shows how to configure an import policy in a RIP network.

- [Requirements on page 47](#)
- [Overview on page 47](#)
- [Configuration on page 48](#)
- [Verification on page 51](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

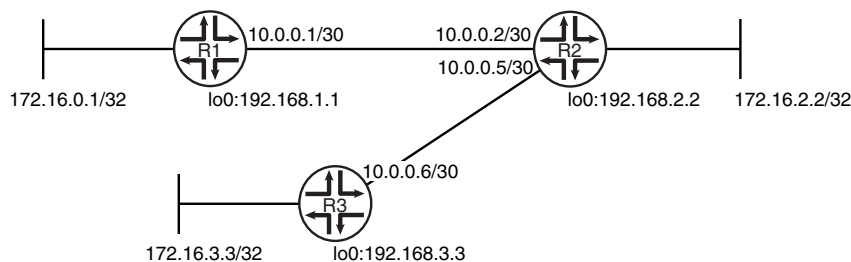
Overview

In this example, Device R1 has an import policy that accepts the 10/8 and 192.168/16 RIP routes and rejects all other RIP routes. This means that the 172.16/16 RIP routes are excluded from Device R1's routing table.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

[Figure 10 on page 48](#) shows the topology used in this example.

Figure 10: RIP Import Policy Network Topology



"CLI Quick Configuration" on page 48 shows the configuration for all of the devices in Figure 10 on page 48. The section "Step-by-Step Procedure" on page 49 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip import rip-import
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set policy-options policy-statement rip-import term 1 from protocol rip
set policy-options policy-statement rip-import term 1 from route-filter 10.0.0.0/8 orlonger
set policy-options policy-statement rip-import term 1 from route-filter 192.168.0.0/16
  orlonger
set policy-options policy-statement rip-import term 1 then accept
set policy-options policy-statement rip-import term 2 then reject

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32

```



```

set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To configure a RIP import policy:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

```

```

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled.

You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip

```

5. Configure the import policy.

```

[edit policy-options policy-statement rip-import]
user@R1# set term 1 from protocol rip
user@R1# set term 1 from route-filter 10.0.0.0/8 orlonger
user@R1# set term 1 from route-filter 192.168.0.0/16 orlonger
user@R1# set term 1 then accept
user@R1# set term 2 then reject

```

6. Apply the import policy.

```
[edit protocols rip]
user@R1# set import rip-import
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  import rip-import;
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}
policy-statement rip-import {
  term 1 {
    from {
      protocol rip;
      route-filter 10.0.0.0/8 orlonger;
      route-filter 192.168.0.0/16 orlonger;
    }
    then accept;
  }
  term 2 {
    then reject;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Looking at the Routes That Device R2 Is Advertising to Device R1 on page 51](#)
- [Looking at the Routes That Device R1 Is Receiving from Device R2 on page 51](#)
- [Checking the Routing Table on page 51](#)
- [Testing the Import Policy on page 52](#)

Looking at the Routes That Device R2 Is Advertising to Device R1

Purpose Verify that Device R2 is sending the expected routes.

Action From operational mode, enter the **show route advertising-protocol rip** command.

```
user@R2> show route advertising-protocol rip 10.0.0.2
```

```
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.0.0.4/30      *[Direct/0] 2d 01:17:44
                  > via fe-1/2/0.5
172.16.2.2/32    *[Direct/0] 2d 04:09:52
                  > via lo0.2
172.16.3.3/32    *[RIP/100] 23:40:02, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
192.168.2.2/32   *[Direct/0] 2d 04:09:52
                  > via lo0.2
192.168.3.3/32   *[RIP/100] 23:40:02, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
```

Meaning Device R2 is sending 172.16/16 routes to Device R1.

Looking at the Routes That Device R1 Is Receiving from Device R2

Purpose Verify that Device R1 is receiving the expected routes.

Action From operational mode, enter the **show route receive-protocol rip** command.

```
user@R1> show route receive-protocol rip 10.0.0.2
```

```
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.0.0.4/30      *[RIP/100] 01:06:03, metric 2, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32    *[RIP/100] 01:06:03, metric 2, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32    *[RIP/100] 01:06:03, metric 3, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1
```

Meaning The output shows that the 172.16/16 routes are excluded.

Checking the Routing Table

Purpose Verify that the routing table is populated with the expected routes.

Action From operational mode, enter the **show route protocol rip** command.

```
user@R1> show route protocol rip
```

```
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[RIP/100] 00:54:34, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32  *[RIP/100] 00:54:34, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32  *[RIP/100] 00:54:34, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
224.0.0.9/32    *[RIP/100] 00:49:00, metric 1
                 MultiRecv
```

Meaning The output shows that the routes have been learned from Device R2 and Device R3.

If you delete or deactivate the import policy, the routing table contains the 172.16/16 routes.

Testing the Import Policy

Purpose By using the **test policy** command, monitor the number of rejected prefixes.

Action From operational mode, enter the **test policy rip-import 172.16/16** command.

```
user@R1> test policy rip-import 172.16/16
Policy rip-import: 0 prefix accepted, 1 prefix rejected
```

Meaning The output shows that the policy rejected one prefix.

Related Documentation

- [Example: Configuring RIP on page 13](#)

Examples: Controlling Traffic with Metrics in a RIP Network

- [Understanding Traffic Control with Metrics in a RIP Network on page 52](#)
- [Example: Controlling Traffic in a RIP Network with an Incoming Metric on page 53](#)
- [Example: Controlling Traffic in a RIP Network with an Outgoing Metric on page 55](#)
- [Example: Configuring the Metric Value Added to Imported RIP Routes on page 56](#)

Understanding Traffic Control with Metrics in a RIP Network

To tune a RIP network and to control traffic flowing through the network, you increase or decrease the cost of the paths through the network. RIP provides two ways to modify the path cost: an incoming metric and an outgoing metric, which are each set to 1 by default. In other words, by default, the metric of routes that RIP imports from a neighbor or exports to a neighbor is incremented by 1. These routes include those learned from RIP as well as those learned from other protocols. The metrics are attributes that specify the cost of any route advertised through a host. By increasing or decreasing the

metrics—and thus the cost—of links throughout the network, you can control packet transmission across the network.

The incoming metric modifies the cost of an individual segment when a route across the segment is imported into the routing table. For example, if you set the incoming metric on the segment to **3**, the individual segment cost along the link is changed from 1 to **3**. The increased cost affects all route calculations through that link. Other routes that were previously excluded because of a high hop count might now be included in the router's forwarding table.

The outgoing metric modifies the path cost for all the routes advertised out of a particular interface. Unlike the incoming metric, the outgoing metric modifies the routes that other routers are learning and thereby controls the way they send traffic.

If an exported route was learned from a member of the same RIP group, the metric associated with that route is the normal RIP metric. For example, a RIP route with a metric of 5 learned from a neighbor configured with an incoming metric of 2 is advertised with a combined metric of 7 when advertised to neighbors in the same group. However, if this route was learned from a RIP neighbor in a different group or from a different protocol, the route is advertised with the metric value configured in the outgoing metric for that group.

You might want to increase the metric of routes to decrease the likelihood that a particular route is selected and installed in the routing table. This process is sometimes referred to as *route poisoning*. Some reasons that you might want to poison a route are that the route is relatively expensive to use, or it has relatively low bandwidth.

A route with a higher metric than another route becomes the active route only when the lower-metric route becomes unavailable. In this way, the higher-metric route serves as a backup path.

One way to increase the metric of imported routes is to configure an import policy. Another way is to include the **metric-in** statement in the RIP neighbor configuration. One way to increase the metric of export routes is to configure an export policy. Another way is to include the **metric-out** statement in the RIP neighbor configuration.

Example: Controlling Traffic in a RIP Network with an Incoming Metric

This example shows how to control traffic with an incoming metric.

- [Requirements on page 53](#)
- [Overview on page 54](#)
- [Configuration on page 54](#)
- [Verification on page 55](#)

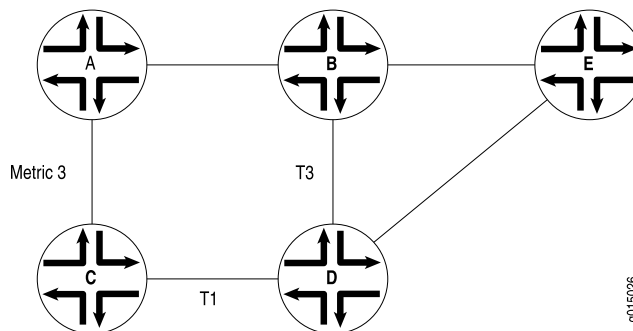
Requirements

Before you begin, define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through the RIP routing exchanges. See [“Example: Configuring a Basic RIP Network” on page 14](#).

Overview

In this example, routes to Router D are received by Router A across both of its RIP-enabled interfaces as shown in Figure 11 on page 54. Because the route through Router B and the route through Router C have the same number of hops, both routes are imported into the forwarding table. However, because the T3 link from Router B to Router D has a higher bandwidth than the T1 link from Router C to Router D, you want traffic to flow from Router A through Router B to Router D.

Figure 11: Controlling Traffic in a RIP Network with the Incoming Metric



To force this flow, you can modify the route metrics as they are imported into Router A's routing table. By setting the incoming metric on the interface from Router A to Router C, you modify the metric on all routes received through that interface. Setting the incoming route metric on Router A changes only the routes in Router A's routing table, and affects only how Router A sends traffic to Router D. Router D's route selection is based on its own routing table, which, by default, includes no adjusted metric values.

In the example, Router C receives a route advertisement from Router D and readvertises the route to Router A. When Router A receives the route, it applies the incoming metric on the interface. Instead of incrementing the metric by 1 (the default), Router A increments it by 3 (the configured incoming metric), giving the route from Router A to Router D through Router C a total path metric of 4. Because the route through Router B has a metric of 2, it becomes the preferred route for all traffic from Router A to Router D.

This example uses a RIP group called **alpha 1** on interface **ge-0/0/0**.

Configuration

Step-by-Step Procedure

To control traffic with an incoming metric:

1. Enable RIP on the interface.

```
[edit protocols rip]
user@host# set group alpha1 neighbor ge-0/0/0
```
2. Set the incoming metric.

```
[edit protocols rip]
user@host# set metric-in 3
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
```

```
user@host# commit
```

Verification

To verify that the configuration is working properly, enter the **show route protocols rip** command.

Example: Controlling Traffic in a RIP Network with an Outgoing Metric

This example shows how to control traffic with an outgoing metric.

- [Requirements on page 55](#)
- [Overview on page 55](#)
- [Configuration on page 56](#)
- [Verification on page 56](#)

Requirements

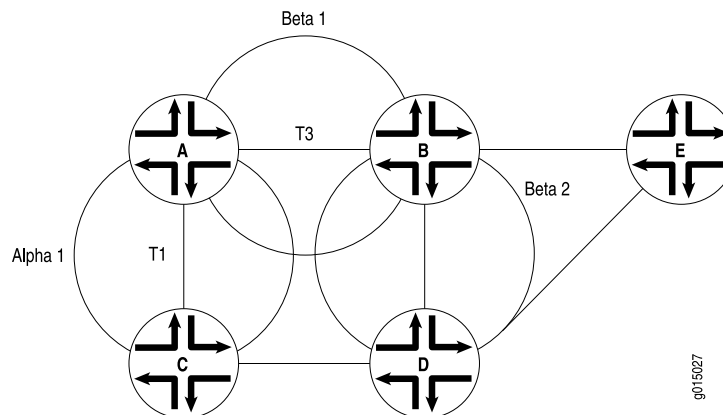
Before you begin:

- Define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through RIP routing exchanges. See [“Example: Configuring a Basic RIP Network” on page 14](#).
- Control traffic with an incoming metric. See [“Example: Controlling Traffic in a RIP Network with an Incoming Metric” on page 53](#).

Overview

In this example, each route from Router A to Router D has two hops as shown in [Figure 12 on page 55](#). However, because the link from Router A to Router B in the RIP group has a higher bandwidth than the link from Router A to Router C in RIP group Alpha 1, you want traffic from Router D to Router A to flow through Router B. To control the way Router D sends traffic to Router A, you can alter the routes that Router D receives by configuring the outgoing metric on Router A's interfaces in the Alpha 1 RIP group.

Figure 12: Controlling Traffic in a RIP Network with the Outgoing Metric



If the outgoing metric for the Alpha 1 RIP group—the A-to-C link—is changed to 3, Router D calculates the total path metric from Router A through Router C as 4. In contrast, the unchanged default total path metric to Router A through Router B in the RIP group is 2. The fact that Router A's interfaces belong to two different RIP groups allows you to configure two different outgoing metrics on its interfaces, because you configure path metrics at the group level.

By configuring the outgoing metric, you control the way Router A sends traffic to Router D. By configuring the outgoing metric on the same router, you control the way Router D sends traffic to Router A.

This example uses an outgoing metric of 3.

Configuration

Step-by-Step Procedure

To control traffic with an outgoing metric:

1. Set the outgoing metric.

```
[edit protocols rip group alpha1]  
user@host# set metric-out 3
```
2. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify that the configuration is working properly, enter the **show protocols rip** command.

Example: Configuring the Metric Value Added to Imported RIP Routes

This example shows how to change the default metric to be added to incoming routes to control the route selection process.

- [Requirements on page 56](#)
- [Overview on page 56](#)
- [Configuration on page 57](#)
- [Verification on page 60](#)

Requirements

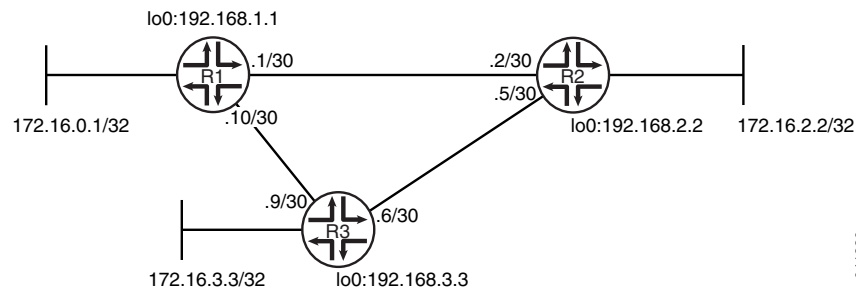
No special configuration beyond device initialization is required before configuring this example.

Overview

Normally, when multiple routes are available, RIP selects the route with the lowest hop count. Changing the default metric enables you to control the route selection process such that a route with a higher hop count can be preferred over of a route with a lower hop count.

[Figure 13 on page 57](#) shows the topology used in this example.

Figure 13: RIP Incoming Metrics Network Topology



Device R1 has two potential paths to reach 172.16.2.2/32. The default behavior is to send traffic out the 0.1/30 interface facing Device R2. Suppose, though, that the path through Device R3 is less expensive to use or has higher bandwidth links. This example shows how to use the **metric-in** statement to ensure that Device R1 uses the path through Device R3 to reach 172.16.2.2/32. “[CLI Quick Configuration](#)” on page 57 shows the configuration for all of the devices in Figure 13 on page 57. The section “[Step-by-Step Procedure](#)” on page 58 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 description to-R2
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces ge-1/2/1 unit 10 description to-R3
set interfaces ge-1/2/1 unit 10 family inet address 10.0.0.10/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group primary export advertise-routes-through-rip
set protocols rip group primary neighbor ge-1/2/1.10
set protocols rip group secondary export advertise-routes-through-rip
set protocols rip group secondary neighbor fe-1/2/0.1 metric-in 4
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces ge-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor ge-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces ge-1/2/1 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group neighbor ge-1/2/1.9
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To configure a RIP metrics:

1. Configure the network interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 description to-R2
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R1# set ge-1/2/1 unit 10 description to-R3
user@R1# set ge-1/2/1 unit 10 family inet address 10.0.0.10/30

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP groups and add the interfaces.

To configure RIP in Junos OS, you must configure one or more groups that contain the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

For the interface that is facing Device R2, the **metric-in 4** setting causes this route to be less likely to be chosen as the active route.

```
[edit protocols rip]
user@R1# set group primary neighbor ge-1/2/1.10
user@R1# set group secondary neighbor fe-1/2/0.1 metric-in 4
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip]
user@R1# set group primary export advertise-routes-through-rip
user@R1# set group secondary export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    description to-R2;
    family inet {
      address 10.0.0.1/30;
    }
  }
}
ge-1/2/1 {
  unit 10 {
    description to-R3;
    family inet {
      address 10.0.0.10/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  group primary {
    export advertise-routes-through-rip;
    neighbor ge-1/2/1.10;
  }
  group secondary {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1 {
      metric-in 4;
    }
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the Expected Route Is Active on page 60](#)
- [Removing the metric-in Statement on page 60](#)

Verifying That the Expected Route Is Active

Purpose Make sure that to reach 172.16.2.2/32, Device R1 uses the path through Device R3.

Action From operational mode, enter the **show route 172.16.2.2** command.

```
user@R1> show route 172.16.2.2
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.2.2/32      *[RIP/100] 00:15:46, metric 3, tag 0
                   > to 10.0.0.9 via ge-1/2/1.10
```

Meaning The **to 10.0.0.9 via ge-1/2/1.10** output shows that Device R1 uses the path through Device R3 to reach 172.16.2.2/32. The metric for this route is 3.

Removing the metric-in Statement

Purpose Delete or deactivate the **metric-in** statement to see what happens to the 172.16.2.2/32 route.

Action 1. From configuration mode, deactivate the **metric-in** statement.

```
[edit protocols rip group secondary neighbor fe-1/2/0.1]
user@R1# deactivate metric-in
user@R1# commit
```

2. From operational mode, enter the **show route 172.16.2.2** command.

```
user@R1> show route 172.16.2.2
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.2.2/32      *[RIP/100] 00:00:06, metric 2, tag 0
                   > to 10.0.0.2 via fe-1/2/0.1
```

Meaning The **to 10.0.0.2 via fe-1/2/0.1** output shows that Device R1 uses the path through Device R2 to reach 172.16.2.2/32. The metric for this route is 2.

Related Documentation

- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 47](#)

Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets

- [Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets on page 61](#)
- [Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 61](#)

Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets

RIP version 1 (RIPv1) and RIP version 2 (RIPv2) can run simultaneously. This might make sense when you are migrating a RIPv1 network to a RIPv2 network. This also allows interoperability with a device that supports RIPv1 but not RIPv2.

By default, when RIP is enabled on an interface, Junos OS receives both RIPv1 and RIPv2 packets and sends only RIPv2 packets. You can configure this behavior by including the `send` and `receive` statements in the RIP configuration.

Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets

This example shows how to configure whether the RIP update messages conform to RIP version 1 (RIPv1) only, to RIP version 2 (RIPv2) only, or to both versions. You can also disable the sending or receiving of update messages.

- [Requirements on page 61](#)
- [Overview on page 61](#)
- [Configuration on page 62](#)
- [Verification on page 64](#)

Requirements

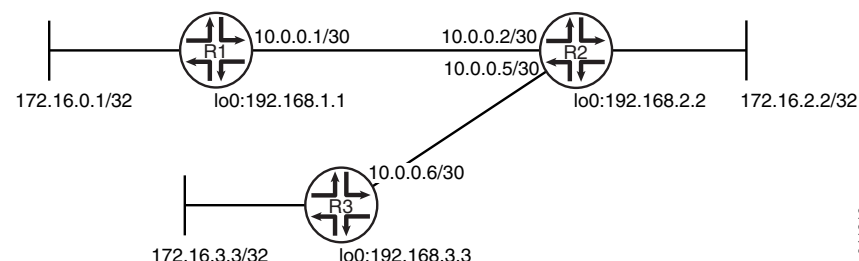
No special configuration beyond device initialization is required before configuring this example.

Overview

By default, when RIP is enabled on an interface, Junos OS receives both RIPv1 and RIPv2 packets and sends only RIPv2 packets.

[Figure 14 on page 61](#) shows the topology used in this example.

Figure 14: Sending and Receiving RIPv1 and RIPv2 Packets Network Topology



In this example, Device R1 is configured to receive only RIPv2 packets.

“[CLI Quick Configuration](#)” on [page 62](#) shows the configuration for all of the devices in [Figure 14 on page 61](#). The section “[Step-by-Step Procedure](#)” on [page 62](#) describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1 receive version-2
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure a RIP packet versions that can be received:

1. Configure the network interfaces.

```
[edit interfaces]
```

```
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

```
user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
```

```
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP groups and add the interfaces.

To configure RIP in Junos OS, you must configure one or more groups that contain the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

For the interface that is facing Device R2, the **receive version-2** setting causes this interface to accept only RIPv2 packets.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1 receive version-2
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1 {
      receive version-2;
    }
  }
}
```

```

}
user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying That the Receive Mode Is Set to RIPv2 Only

Purpose Make sure that the interfacing Device R2 is configured to receive only RIPv2 packets, instead of both RIPv1 and RIPv2 packets.

Action From operational mode, enter the **show rip neighbor** command.

```
user@R1> show rip neighbor
```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-1/2/0.1	Up	10.0.0.1	224.0.0.9	mcast	v2 only	1

Meaning In the output, the **Receive Mode** field displays **v2 only**. The default **Receive Mode** is **both**.

Related Documentation

- [Example: Configuring RIP on page 13](#)

Example: Redistributing Routes Among RIP Instances

- [Understanding Route Redistribution Among RIP instances on page 64](#)
- [Example: Redistributing Routes Between Two RIP Instances on page 65](#)

Understanding Route Redistribution Among RIP instances

You can redistribute routes among RIP processes. Another way to say this is to export RIP routes from one RIP instance to other RIP instances.

In Junos OS, route redistribution among routing instances is accomplished by using routing table groups, also called RIB groups. Routing table groups allow you to import and export routes from a protocol within one routing table into another routing table.



NOTE: In contrast, the policy-based import and export functions allow you import and export routes between different protocols within the same routing table.

Consider the following partial example:

```
protocols {
  rip {
    rib-group inet-to-voice;
  }
}
routing-instances {
  voice {
    protocols {
      rip {
        rib-group voice-to-inet;
      }
    }
  }
}
routing-options {
  rib-groups {
    inet-to-voice {
      import-rib [ inet.0 voice.inet.0 ];
    }
    voice-to-inet {
      import-rib [ voice.inet.0 inet.0 ];
    }
  }
}
```

The way to read the **import-rib** statement is as follows. Take the routes from the protocol (RIP, in this case), and import them into the primary (or local) routing table and also into any other routing tables listed after this. The primary routing table is the routing table where the routing table group is being used. That would be either **inet.0** if used in the main routing instance or **voice.inet.0** if used within the routing instance. In the **inet-to-voice** routing table group, **inet.0** is listed first because this routing table group is used in the main routing instance. In the **voice-to-inet** routing table group, **voice.inet.0** is listed first because this routing table group is used in the voice routing instance.

Example: Redistributing Routes Between Two RIP Instances

This example shows how to configure a RIP routing instance and control the redistribution of RIP routes between the routing instance and the master instance.

- [Requirements on page 65](#)
- [Overview on page 66](#)
- [Configuration on page 66](#)
- [Verification on page 69](#)

Requirements

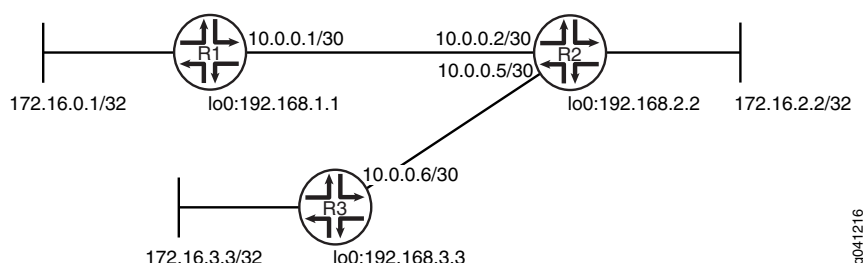
No special configuration beyond device initialization is required before configuring this example.

Overview

When you create a routing instance called **voice**, Junos OS creates a routing table called **voice.inet.0**. The example shows how to install routes learned through the master RIP instance into the **voice.inet.0** routing table. The example also shows how to install routes learned through the voice routing instance into **inet.0**. This is done by configuring routing table groups. RIP routes are installed into each routing table that belongs to a routing table group.

Figure 15 on page 66 shows the topology used in this example.

Figure 15: Redistributing Routes Between RIP Instances Network Topology



"CLI Quick Configuration" on page 66 shows the configuration for all of the devices in Figure 15 on page 66. The section "Step-by-Step Procedure" on page 67 describes the steps on Device R2.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```

Device R1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group to-R2 export advertise-routes-through-rip
set protocols rip group to-R2 neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

Device R2
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip rib-group inet-to-voice
set protocols rip group to-R3 export advertise-routes-through-rip
set protocols rip group to-R3 neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct

```

```

set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set routing-instances voice protocols rip group to-R1 export advertise-routes-through-rip
set routing-instances voice interface fe-1/2/0.2
set routing-instances voice protocols rip rib-group voice-to-inet
set routing-instances voice protocols rip group to-R1 neighbor fe-1/2/0.2
set routing-options rib-groups inet-to-voice import-rib inet.0
set routing-options rib-groups inet-to-voice import-rib voice.inet.0
set routing-options rib-groups voice-to-inet import-rib voice.inet.0
set routing-options rib-groups voice-to-inet import-rib inet.0

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group to-R2 export advertise-routes-through-rip
set protocols rip group to-R2 neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To redistribute RIP routes between routing instances:

1. Configure the network interfaces.

```

[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 5 family inet address 10.0.0.5/30

```

```

user@R2# set lo0 unit 2 family inet address 192.168.2.2/32
user@R2# set lo0 unit 2 family inet address 172.16.2.2/32

```

2. Create the routing instance, and add one or more interfaces to the routing instance.

```

[edit routing-instances voice]
user@R2# set interface fe-1/2/0.2

```

3. Create the RIP groups and add the interfaces.

```

[edit protocols rip group to-R3]
user@R2# set neighbor fe-1/2/1.5

[edit routing-instances voice protocols rip group to-R1]
user@R2# set neighbor fe-1/2/0.2

```

4. Create the routing table groups.

```

[edit routing-options rib-groups]
user@R2# set inet-to-voice import-rib inet.0
user@R2# set inet-to-voice import-rib voice.inet.0

```

```
user@R2# set voice-to-inet import-rib voice.inet.0
user@R2# set voice-to-inet import-rib inet.0
```

5. Apply the routing table groups.

```
[edit protocols rip]
user@R2# set rib-group inet-to-voice
```

```
[edit routing-instances voice protocols rip]
user@R2# set rib-group voice-to-inet
```

6. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R2# set from protocol direct
user@R2# set from protocol rip
user@R2# set then accept
```

7. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group to-R3]
user@R2# set export advertise-routes-through-rip
```

```
[edit routing-instances voice protocols rip group to-R1]
user@R2# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R2# show interfaces
fe-1/2/0 {
  unit 2 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 5 {
    family inet {
      address 10.0.0.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.2.2/32;
      address 172.16.2.2/32;
    }
  }
}
```

```

user@R2# show protocols
rip {
  rib-group inet-to-voice;
  group to-R3 {
    export advertise-routes-through-rip;
    neighbor fe-1/2/1.5;
  }
}

user@R2# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

user@R2# show routing-instances
voice {
  interface fe-1/2/0.2;
  protocols {
    rip {
      rib-group voice-to-inet;
      group to-R1 {
        export advertise-routes-through-rip;
        neighbor fe-1/2/0.2;
      }
    }
  }
}

user@R2# show routing-options
rib-groups {
  inet-to-voice {
    import-rib [ inet.0 voice.inet.0 ];
  }
  voice-to-inet {
    import-rib [ voice.inet.0 inet.0 ];
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Routing Tables

Purpose Make sure that the routing tables contain the expected routes.

Action From operational mode, enter the **show route protocol rip** command.

```

user@R2> show route protocol rip
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.1/32      *[RIP/100] 01:58:14, metric 2, tag 0

```

```
172.16.3.3/32      > to 10.0.0.1 via fe-1/2/0.2
                  *[RIP/100] 02:06:03, metric 2, tag 0
192.168.1.1/32    > to 10.0.0.6 via fe-1/2/0.5
                  *[RIP/100] 01:58:14, metric 2, tag 0
192.168.3.3/32    > to 10.0.0.1 via fe-1/2/0.2
                  *[RIP/100] 02:06:03, metric 2, tag 0
224.0.0.9/32      > to 10.0.0.6 via fe-1/2/0.5
                  *[RIP/100] 01:44:13, metric 1
                  MultiRecv

voice.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.1/32      *[RIP/100] 02:06:03, metric 2, tag 0
                  > to 10.0.0.1 via fe-1/2/0.2
172.16.3.3/32      *[RIP/100] 01:58:14, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
192.168.1.1/32     *[RIP/100] 02:06:03, metric 2, tag 0
                  > to 10.0.0.1 via fe-1/2/0.2
192.168.3.3/32     *[RIP/100] 01:58:14, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
224.0.0.9/32       *[RIP/100] 01:44:13, metric 1
                  MultiRecv
```

Meaning The output shows that both routing tables contain all of the RIP routes.

Related Documentation

- [Example: Configuring RIP on page 13](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 47](#)

Example: Configuring RIP Timers

- [Understanding RIP Timers on page 70](#)
- [Example: Configuring RIP Timers on page 71](#)

Understanding RIP Timers

RIP uses several timers to regulate its operation.

The update interval is the interval at which routes that are learned by RIP are advertised to neighbors. This timer controls the interval between routing updates. The update interval is set to 30 seconds, by default, with a small random amount of time added when the timer is reset. This added time prevents congestion that can occur if all routing devices update their neighbors simultaneously.

To configure the update time interval, include the **update-interval** statement:

```
update-interval seconds;
```

seconds can be a value from 10 through 60.

You can set a route timeout interval. If a route is not refreshed after being installed in the routing table by the specified time interval, the route is marked as invalid and is removed from the routing table after the hold-down period expires.

To configure the route timeout for RIP, include the **route-timeout** statement:

route-timeout *seconds*;

seconds can be a value from 30 through 360. The default value is 180 seconds.

RIP routes expire when either a route timeout limit is met or a route metric reaches infinity, and the route is no longer valid. However, the expired route is retained in the routing table for a specified period so that neighbors can be notified that the route has been dropped. This time period is set by configuring the hold-down timer. Upon expiration of the hold-down timer, the route is removed from the routing table.

To configure the hold-down timer for RIP, include the **holddown** statement:

holddown *seconds*;

seconds can be a value from 10 through 180. The default value is 120 seconds.



NOTE: In Junos OS Release 11.1 and later, a retransmission timer is available for RIP demand circuits.

Generally, we recommend against changing the RIP timers, unless the effects of a change are well understood. The route timeout should be at least three times the update interval. The hold-down timer must be greater than the route timeout. Normally, the default values are best left in effect for standard operations.

Example: Configuring RIP Timers

This example shows how to configure the RIP update interval and how to monitor the impact of the change.

- [Requirements on page 71](#)
- [Overview on page 71](#)
- [Configuration on page 72](#)
- [Verification on page 74](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

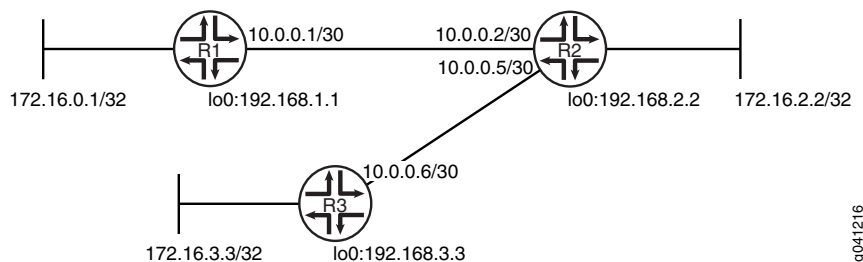
In this example, Device R2 has an update interval of 60 seconds for its neighbor, Device R1, and an update interval of 10 seconds for its neighbor, Device R3.

This example is not necessarily practical, but it is shown for demonstration purposes. Generally, we recommend against changing the RIP timers, unless the effects of a change are well understood. Normally, the default values are best left in effect for standard operations.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

Figure 16 on page 72 shows the topology used in this example.

Figure 16: RIP Timers Network Topology



"CLI Quick Configuration" on page 72 shows the configuration for all of the devices in Figure 16 on page 72. The section "Step-by-Step Procedure" on page 73 describes the steps on Device R2.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set interfaces lo0 unit 1 family inet address 172.16.0.1/32 set interfaces lo0 unit 1 family inet address 192.168.1.1/32 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set interfaces lo0 unit 2 family inet address 192.168.2.2/32 set interfaces lo0 unit 2 family inet address 172.16.2.2/32 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 update-interval 60 set protocols rip group rip-group neighbor fe-1/2/1.5 update-interval 10 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R3	<pre> set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30 set interfaces lo0 unit 3 family inet address 192.168.3.3/32 set interfaces lo0 unit 3 family inet address 172.16.3.3/32 set protocols rip group rip-group export advertise-routes-through-rip </pre>


```

set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the RIP update interval:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 5 family inet address 10.0.0.5/30

```

```

user@R2# set lo0 unit 2 family inet address 192.168.2.2/32
user@R2# set lo0 unit 2 family inet address 172.16.2.2/32

```

2. Configure different update intervals for the two RIP neighbors.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R2# set neighbor fe-1/2/0.2 update-interval 60
user@R2# set neighbor fe-1/2/1.5 update-interval 10

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R2# set from protocol direct
user@R2# set from protocol rip
user@R2# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R2# set export advertise-routes-through-rip

```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R2# show interfaces
fe-1/2/0 {
  unit 2 {

```

```
        family inet {
            address 10.0.0.2/30;
        }
    }
}
fe-1/2/1 {
    unit 5 {
        family inet {
            address 10.0.0.5/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.2.2/32;
            address 172.16.2.2/32;
        }
    }
}

user@R2# show protocols
rip {
    group rip-group {
        export advertise-routes-through-rip;
        neighbor fe-1/2/0.2 {
            update-interval 60;
        }
        neighbor fe-1/2/1.5 {
            update-interval 10;
        }
    }
}

user@R2# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking the RIP Updates Sent by Device R2 on page 74](#)
- [Checking the RIP Updates Received by Device R2 on page 75](#)
- [Checking the RIP Updates Received by Device R3 on page 76](#)

Checking the RIP Updates Sent by Device R2

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```
user@R2> show rip statistics
```

```
RIPv2 info: port 520; holddown 120s.
```

```
    rts learned  rts held down  rqsts dropped  resps dropped
           4             2             0             0
```

```
fe-1/2/0.2: 2 routes learned; 5 routes advertised; timeout 180s; update interval 60s
```

Counter	Total	Last 5 min	Last minute
Updates Sent	123	5	1
Triggered Updates Sent	0	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	244	10	2
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

```
fe-1/2/1.5: 2 routes learned; 5 routes advertised; timeout 180s; update interval 10s
```

Counter	Total	Last 5 min	Last minute
Updates Sent	734	32	6
Triggered Updates Sent	0	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	245	11	2
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **update interval** field shows that the interval is 60 seconds for Neighbor R1 and 10 seconds for Neighbor R3. The **Updates Sent** field shows that Device R2 is sending updates to Device R1 at roughly 1/6 of the rate that it is sending updates to Device R3.

Checking the RIP Updates Received by Device R2

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```
user@R1> show rip statistics
```

```
RIPv2 info: port 520; holddown 120s.
```

```
    rts learned  rts held down  rqsts dropped  resps dropped
           5             0             0             0
```

```
fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval 30s
```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	312	10	2
Triggered Updates Sent	2	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	181	5	1
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	1	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **RIPv2 Updates Received** field shows the number of updates received from Device R2.

Checking the RIP Updates Received by Device R3

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```
user@R3> show rip statistics
```

```
RIPv2 info: port 520; holddown 120s.
```

```
    rts learned  rts held down  rqsts dropped  resps dropped
          5             0             0             0
```

```
fe-1/2/0.6: 5 routes learned; 2 routes advertised; timeout 180s; update interval 30s
```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	314	11	2
Triggered Updates Sent	1	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	827	31	6
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **RIPv2 Updates Received** field shows the number of updates received from Device R2.

- Related Documentation**
- [Example: Configuring RIP on page 13](#)
 - [Example: Configuring RIP Demand Circuits on page 77](#)

Example: Configuring RIP Demand Circuits

- [RIP Demand Circuits Overview on page 77](#)
- [Example: Configuring RIP Demand Circuits on page 79](#)

RIP Demand Circuits Overview

RIP periodically sends routing information (RIP packets) to neighboring devices. These periodic broadcasts can consume bandwidth resources and interfere with network traffic by preventing WAN circuits from being closed. Demand circuits for RIP is defined in RFC 2091 and overcomes these issues by exchanging incremental updates on demand.

A demand circuit is a point-to-point connection between two neighboring interfaces configured for RIP. Demand circuits preserve bandwidth by establishing a link when data needs to be transferred, and terminating the link when the data transfer is complete. Demand circuits increase the efficiency of RIP on the configured interfaces by offering minimal network overhead in terms of messages passed between the demand circuit end points, thus conserving resources and reducing costs.

By configuring RIP demand circuits, a specific event triggers the device to send an update, thereby eliminating the periodic transmission of RIP packets over the neighboring interface. To save overhead, the device sends RIP information only when changes occur in the routing database, such as:

- The device is first powered on
- The device receives a request for route update information
- A change occurs in the network
- The demand circuit goes down or comes up

The device sends update requests, update responses, and acknowledgments. In addition, the device retransmits updates and requests until valid acknowledgments are received. The device dynamically learns RIP neighbors. If the neighboring interface goes down, RIP flushes routes learned from the neighbor's IP address.

Routes learned from demand circuits do not age like other RIP entries because demand circuits are in a permanent state. Routes in a permanent state are only removed under the following conditions:

- A formerly reachable route changes to unreachable in an incoming response
- The demand circuit is down due to an excessive number of unacknowledged retransmissions

You can also set the RIP hold-down timer and the RIP demand circuit retransmission timer to regulate performance. The demand circuit uses these timers to determine if

there is a change that requires update messages to be sent. There is also a database timer that runs only when RIP flushes learned routes from the routing table.

This topic includes the following sections:

- [RIP Demand Circuit Packets on page 78](#)
- [Timers Used by RIP Demand Circuits on page 79](#)

RIP Demand Circuit Packets

When you configure an interface for RIP demand circuits, the supported command field packet types are different than those for RIP version 1 and RIP version 2. RIP packets for RIP demand circuits contain three additional packet types and an extended 4-byte update header. Both RIP version 1 and RIP version 2 support the three packet types and the extended 4-byte header. [Table 5 on page 78](#) describes the three packet types.

Table 5: RIP Demand Circuit Packet Types

Packet Type	Description
Update Request	Update request messages seek information for the device's routing table. This message is sent when the device is first powered on or when a down demand circuit comes up. The device sends this message every 5 seconds (by default) until an update response message is received.
Update Response	Update response messages are sent in response to an update request message, which occurs when the device is first powered on or when a down demand circuit comes up. Each update response message contains a sequence number that the neighbor uses to acknowledge the update request.
Update Acknowledge	Update acknowledge messages are sent in response to every update response message received by the neighbor.



NOTE: These packets are only valid on interfaces configured for RIP demand circuits. If a demand circuit receives a RIP packet that does not contain these packet types, it silently discards the packet and logs an error message similar to the following:

Ignoring RIP packet with invalid version 0 from neighbor 10.0.0.0 and source 10.0.0.1

Timers Used by RIP Demand Circuits

RIP demand circuits use the RIP hold-down timer and the RIP demand circuit retransmission timer to regulate performance and to determine if there is a change in the network that requires the device to send update messages. The hold-down timer is a global RIP timer that affects the entire RIP configuration. Whatever range you configure for RIP applies to RIP demand circuits. The retransmission timer affects only RIP demand circuits. In addition, there is a database timer that runs only when RIP flushes learned routes from the routing table.

- **Hold-down timer (global RIP timer)**—Use the hold-down timer to configure the number of seconds that RIP waits before updating the routing table. The value of the hold-down timer affects the entire RIP configuration, not just the demand circuit interfaces. The hold-down timer starts when a route timeout limit is met, when a formerly reachable route is unreachable, or when a demand circuit interface is down. When the hold-down timer is running, routes are advertised as unreachable on other interfaces. When the hold-down timer expires, the route is removed from the routing table if all destinations are aware that the route is unreachable or the remaining destinations are down. By default, RIP waits 120 seconds between routing table updates. The range is from 10 to 180 seconds.
- **Retransmission timer (RIP demand circuit timer)**—RIP demand circuits send update messages every 5 seconds to an unresponsive peer. Use the retransmission timer to limit the number of times a demand circuit resends update messages to an unresponsive peer. If the configured retransmission threshold is reached, routes from the next hop router are marked as unreachable and the hold-down timer starts. The value of the retransmission timer affects only the demand circuit interfaces. To determine the number of times to resend the update message, use the following calculation:

$$5 \text{ seconds} \times \text{number of retransmissions} = \text{retransmission seconds}$$

The retransmission range is from 5 through 180 seconds, which corresponds to sending an update message a minimum of 1 time (5 seconds) and a maximum of 36 times (180 seconds).

- **Database timer (global timeout timer)**—Routes learned from demand circuits do not age like other RIP entries because demand circuits are in a permanent state. On a RIP demand circuit, the database timer starts upon receipt of the update response message with the flush flag sent from a RIP demand circuit peer. When the neighbor receives this message, all routes from that peer are flushed, and the database timer starts and runs for the configured route timeout interval. When the database timer is running, routes are still advertised as reachable on other interfaces. When the database timer expires, the device advertises all routes from its peer as unreachable.

Example: Configuring RIP Demand Circuits

This example describes how to configure an interface as a RIP demand circuit.

- [Requirements on page 80](#)
- [Overview on page 80](#)

- [Configuration on page 80](#)
- [Verification on page 82](#)

Requirements

Before you begin, configure the device interfaces. See the Router Interfaces or the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

A demand circuit is a point-to-point connection between two neighboring interfaces configured for RIP. Demand circuits increase the efficiency of RIP on the configured interfaces by eliminating the periodic transmission of RIP packets. Demand circuits preserve bandwidth by establishing a link when data needs to be transferred, and terminating the link when the data transfer is complete. In this example, two devices are connected using SONET/SDH interfaces.



NOTE: When you configure RIP demand circuits, any silent removal of the RIP configuration goes unnoticed by the RIP peer and leads to stale entries in the routing table. To clear the stale entries, deactivate and reactivate RIP on the neighboring devices.

In this example, you configure interface **so-0/1/0** with the following settings:

- **demand-circuit**—Configures the interface as a demand circuit. To complete the demand circuit, you must configure both ends of the pair as demand circuits.
- **max-retrans-time**—RIP demand circuits send update messages every 5 seconds to an unresponsive peer. Use the retransmission timer to limit the number of times a demand circuit resends update messages to an unresponsive peer. If the configured retransmission threshold is reached, routes from the next-hop router are marked as unreachable, and the hold-down timer starts. The value of the retransmission timer affects only the demand circuit interfaces. To determine the number of times to resend the update message, use the following calculation:

$$5 \text{ seconds} \times \text{retransmissions} = \text{retransmission seconds}$$

For example, if you want the demand circuit to send only two update messages to an unresponsive peer, the calculation is: $5 \times 2 = 10$. When you configure the retransmission timer, you enter 10 seconds.

The retransmission range is from 5 through 180 seconds, which corresponds to sending an update message a minimum of 1 time (5 seconds) and a maximum of 36 times (180 seconds).

Configuration

In the following example, you configure a neighboring interface to be a RIP demand circuit and save the configuration.

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

configuration, and then copy and paste the commands in the CLI at the **[edit]** hierarchy level.

```
set interfaces so-0/1/0 unit 0 family inet address 192.0.2.0/24
set protocols rip group group1 neighbor so-0/1/0 demand-circuit
set protocols rip group group1 neighbor so-0/1/0 max-retrans-time 10
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode in the CLI User Guide*.

To configure a RIP demand circuit on one neighboring interface:

1. Configure the interface.

```
[edit interfaces]
user@host# set so-0/1/0 unit 0 family inet address 192.0.2.0/24
```

2. Configure the neighbor as a demand circuit.

```
[edit protocols rip]
user@host# set group group1 neighbor so-0/1/0 demand-circuit
```

3. Configure the demand circuit retransmission timer.

```
[edit protocols rip]
user@host# set group group1 neighbor so-0/1/0 max-retrans-time 10
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```



NOTE: Repeat this entire configuration on the other neighboring interface.

Results

Confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
so-0/1/0 {
  unit 0 {
    family inet {
      address 192.0.2.0/24;
    }
  }
}

user@host# show protocols
rip {
  group group1 {
    neighbor so-0/1/0 {
```

```

        demand-circuit;
        max-retrans-time 10;
    }
}

```

Verification

Confirm that the configuration is working properly.

Verifying a Demand Circuit Configuration

Purpose Verify that the demand circuit configuration is working.

Action To verify that the demand circuit configuration is in effect, use the [show rip neighbor](#) operational mode command.

```
user@host> show rip neighbor
```

Neighbor	State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
so-0/1/0.0(DC)	Up	10.10.10.2	224.0.0.9	mcast	both	1

When you configure demand circuits, the **show rip neighbor** command displays a DC flag next to the neighboring interface configured for demand circuits.



NOTE: If you configure demand circuits at the `[edit protocols rip group group-name neighbor neighbor-name]` hierarchy level, the output shows only the neighboring interface that you specifically configured as a demand circuit. If you configure demand circuits at the `[edit protocols rip group group-name]` hierarchy level, all of the interfaces in the group are configured as demand circuits. Therefore, the output shows all of the interfaces in that group as demand circuits.

Related Documentation

- [Example: Configuring RIP Timers on page 70](#)

Example: Configuring Point-to-Multipoint RIP Networks

- [Configuring Point-to-Multipoint RIP Networks Overview on page 82](#)
- [Example: Configuring Point-to-Multipoint RIP Networks on page 83](#)

Configuring Point-to-Multipoint RIP Networks Overview

A point-to-multipoint RIP network consists of a device having two or more peers on a single interface. All the devices forming a point-to-multipoint connection are placed in a single broadcast domain.

In a RIP network, a device can have a single peer or multiple peers for an interface. However, the demand circuit feature implementation in a RIP network requires the use

of a single RIP peer. When you configure the following statements, a RIP network with demand circuits can also be configured to have multiple peers on an interface:

- Configuring the interface type to be a multipoint interface by using the [interface-type \(Protocols RIP\) p2mp](#) statement.
- Enabling dynamic peer discovery by using the [dynamic-peers](#) statement (SRX Series devices only).



NOTE: Before configuring the [dynamic-peers](#) statement, IPsec must be configured and IPsec tunnels must be set up by configuring IPsec parameters. Without IPsec configuration, the remote peers have to be explicitly configured at the RIP protocol level by using the [peer address](#) statement. See [Configuring Security Associations for IPsec on an ES PIC](#) for more details.

- Configuring peers by using the [peer address](#) statement.

```
[edit]
protocols {
  rip {
    group red {
      neighbor fe-0/1/3 {
        interface-type (Protocols RIP) p2mp;
        peer address; (or use dynamic-peers;)
      }
    }
  }
}
```

The [show rip statistics peer address](#) command can be used to display the RIP statistics at the peer level. The [clear rip statistics peer address](#) command can be used to clear the RIP statistics for a peer. Alternatively, you can use the [show rip statistics peer all](#) and [clear rip statistics peer all](#) command to display and clear RIP statistics for all peers.

Example: Configuring Point-to-Multipoint RIP Networks

This example shows how to configure a point-to-multipoint RIP network.

- [Requirements on page 83](#)
- [Overview on page 84](#)
- [Configuration on page 85](#)
- [Verification on page 87](#)

Requirements

This example uses the following hardware and software components:

- M Series routers, MX Series routers, T Series routers, or SRX Series devices
- Junos OS Release 12.1 or later

Overview

In a RIP network, a device can have a single peer or multiple peers for an interface. However, the demand circuit feature implementation in a RIP network requires the use of a single RIP peer.

When you include the following statements, the demand circuit implementation can have multiple peers for a given RIP neighbor.

- Configuring the interface type to be a multipoint interface by using the [interface-type \(Protocols RIP\) p2mp](#) statement.
- Enabling dynamic peer discovery by using the [dynamic-peers](#) statement (SRX Series devices only).



NOTE: To configure the [dynamic-peers](#) statement, IPsec tunnels must be set up by configuring IPsec parameters. See [Configuring Security Associations for IPsec on an ES PIC](#) for more details.

- Configuring peers by using the [peer address](#) statement.

```
[edit]
protocols {
  rip {
    group red {
      neighbor fe-0/1/3 {
        interface-type (Protocols RIP) p2mp;
        peer address; (or use dynamic-peers;)
      }
    }
  }
}
```

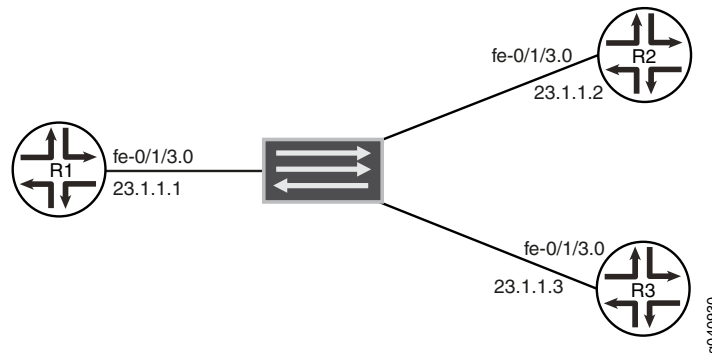
The [show rip statistics peer](#) command can be used to display the RIP statistics at the peer level.

Topology

In this example, Devices R1, R2, and R3 form a point-to-multipoint network. R1 is connected to R2 and to R3 as a point-to-multipoint connection through a switch that places all devices in the same broadcast domain. RIP demand circuits are configured on all three devices. The two peers to R1 are configured statically by using the [peer address](#) statement. The [dynamic-peers](#) statement is not used here.

[Figure 9 on page 43](#) shows the topology used in this example.

Figure 17: Configuring a Point-to-Multipoint RIP Network



Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-0/1/3 unit 0 family inet address 23.1.1.1/24
set policy-options policy-statement accept-rip-routes term from-direct from protocol direct
set policy-options policy-statement accept-rip-routes term from-direct then accept
set policy-options policy-statement accept-rip-routes term from-rip from protocol rip
set policy-options policy-statement accept-rip-routes term from-rip then accept
set protocols rip traceoptions file R1.log size 4m world-readable
set protocols rip traceoptions flag all detail
set protocols rip group red export accept-rip-routes
set protocols rip group red neighbor fe-0/1/3.0 interface-type p2mp
set protocols rip group red neighbor fe-0/1/3.0 peer 23.1.1.2
set protocols rip group red neighbor fe-0/1/3.0 peer 23.1.1.3
set protocols rip group red neighbor fe-0/1/3.0 demand-circuit
set protocols rip group red neighbor fe-0/1/3.0 max-retrans-time 10
  
```

Similarly, configure Devices R2 and R3, omitting the **peer address** configuration statement.

Configuring a Point-to-Multipoint RIP Network (with Demand Circuits)

Step-by-Step Procedure The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the point-to-multipoint feature across a RIP network:

1. Configure the device interface.

```

[edit interfaces fe-0/1/3 unit 0]
user@R1# set family inet address 23.1.1.1/24
  
```
2. Define a policy for exporting RIP routes from the routing table to the protocol for transmission through the network.

```

[edit policy-options policy-statement accept-rip-routes]
user@R1# set term from-direct from protocol direct
  
```

```

user@R1# set term from-direct then accept
user@R1# set term from-rip from protocol rip
user@R1# set term from-rip then accept

```

3. Configure RIP and a RIP group with the defined export policy and point-to-multipoint configuration statements.

```

[edit protocols rip]
user@R1# set traceoptions file R1.log size 4m world-readable
user@R1# set traceoptions flag all detail
user@R1# set group red export accept-rip-routes
user@R1# set group red neighbor fe-0/1/3.0 interface-type p2mp
user@R1# set group red neighbor fe-0/1/3.0 peer 23.1.1.2
user@R1# set group red neighbor fe-0/1/3.0 peer 23.1.1.3
user@R1# set group red neighbor fe-0/1/3.0 demand-circuit
user@R1# set group red neighbor fe-0/1/3.0 max-retrans-time 10

```

Similarly, configure Devices R2 and R3, omitting the **peer address** configuration statement.



NOTE: Configuring **max-retrans-time** is optional. In the absence of this configuration statement, the default retransmission time of 180 seconds is configured.

The configuration used in this example is for a RIP network with demand circuits. To configure RIP for networks without demand circuits, exclude the **demand-circuit** and **max-retrans-time** statements from the configuration and check the resulting output. For more information about configuring RIP demand circuits, see [“Example: Configuring RIP Demand Circuits” on page 79](#).

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, and **show protocols rip** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@R1# show interfaces
fe-0/1/3 {
  unit 0 {
    family inet {
      address 23.1.1.1/24;
    }
  }
}

user@R1# show protocols rip
traceoptions {
  file R1.log size 4m world-readable;
  flag all detail;
}
group red {

```

```
export accept-rip-routes;
neighbor fe-0/1/3.0 {
  interface-type p2mp;
  peer 23.1.1.2;
  peer 23.1.1.3;
  demand-circuit;
  max-retrans-time 10;
}
}

user@R1# show policy-options
policy-statement accept-rip-routes {
  term from-direct {
    from protocol direct;
    then accept;
  }
  term from-rip {
    from protocol rip;
    then accept;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying the Point-to-Multipoint RIP Network

Purpose Verify that the RIP network is functional with the point-to-multipoint feature configured.

Action From operational mode, run the **show rip neighbor** command.

```
user@R1> show rip neighbor
```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-0/1/3.0(DC)	Up	23.1.1.1	23.1.1.2	unicast	unicast	1
fe-0/1/3.0(DC)	Up	23.1.1.1	23.1.1.3	unicast	unicast	1

From operational mode, run the **show rip statistics peer address** command.

```
user@R1> show rip statistics peer 23.1.1.2
```

RIPv2 info: port 520; holddown 120s.

rts learned	rts held down	rqsts dropped	resps dropped
3	0	0	0

fe-0/1/3.0 Peer-Ip 23.1.1.2: 2 routes learned; 3 routes advertised; timeout 180s; update interval 0s

Counter	Total	Last 5 min	Last minute
Updates Sent	0	0	0
Triggered Updates Sent	3	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2	0	0
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	3	0	0

```
user@R1> show rip statistics peer 23.1.1.3
```

RIPv2 info: port 520; holddown 120s.

rts learned	rts held down	rqsts dropped	resps dropped
3	0	0	0

fe-0/1/3.0 Peer-Ip 23.1.1.3: 2 routes learned; 3 routes advertised; timeout 180s; update interval 0s

Counter	Total	Last 5 min	Last minute
Updates Sent	0	0	0
Triggered Updates Sent	3	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2	0	0
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0

none	3	0	0
------	---	---	---

Meaning The RIP network is up and running with the point-to-multipoint feature configured.

Related Documentation

- [Example: Configuring RIP on page 13](#)

Example: Tracing RIP Protocol Traffic

- [Understanding RIP Trace Operations on page 89](#)
- [Example: Tracing RIP Protocol Traffic on page 90](#)

Understanding RIP Trace Operations

You can trace various types of RIP protocol traffic to help debug RIP protocol issues.

To trace RIP protocol traffic, include the **traceoptions** statement at the **[edit protocols rip]** hierarchy level:

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

You can specify the following RIP protocol-specific trace options using the **flag** statement:

- **auth**—RIP authentication
- **error**—RIP error packets
- **expiration**—RIP route expiration processing
- **holddown**—RIP hold-down processing
- **nsr-synchronization**—Nonstop active routing synchronization events
- **packets**—All RIP packets
- **request**—RIP information packets
- **trigger**—RIP triggered updates
- **update**—RIP update packets

You can optionally specify one or more of the following flag modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted



NOTE: Use the **detail** flag modifier with caution as this may cause the CPU to become very busy.

Global tracing options are inherited from the configuration set by the **traceoptions** statement at the **[edit routing-options]** hierarchy level. You can override the following global trace options for the RIP protocol using the **traceoptions flag** statement included at the **[edit protocols rip]** hierarchy level:

- **all**—All tracing operations
- **general**—All normal operations and routing table changes (a combination of the normal and route trace operations)
- **normal**—Normal events
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing



NOTE: Use the trace flag **all** with caution because this may cause the CPU to become very busy.

Example: Tracing RIP Protocol Traffic

This example shows how to trace RIP protocol operations.

- [Requirements on page 90](#)
- [Overview on page 90](#)
- [Configuration on page 91](#)
- [Verification on page 93](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

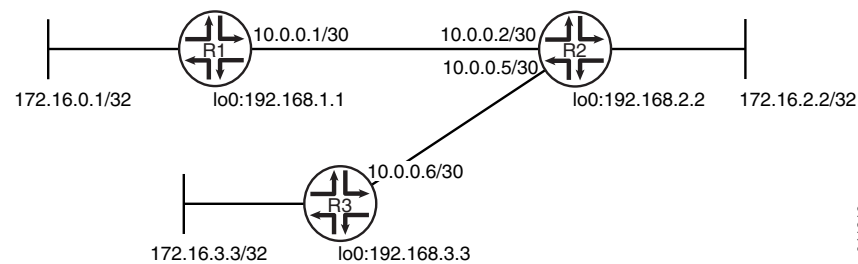
Overview

In this example, Device R1 is set to trace routing information updates.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

[Figure 18 on page 91](#) shows the topology used in this example.

Figure 18: RIP Trace Operations Network Topology



“CLI Quick Configuration” on page 91 shows the configuration for all of the devices in Figure 18 on page 91. The section “Step-by-Step Procedure” on page 92 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip traceoptions file rip-trace-file
set protocols rip traceoptions flag route
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct

```

```
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

**Step-by-Step
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the RIP update interval:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

```
user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Configure the RIP group, and add the interface to the group.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Configure RIP tracing operations.

```
[edit protocols rip traceoptions]
user@R1# set file rip-trace-file
user@R1# set flag route
```

4. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

5. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
```

```

        address 10.0.0.1/30;
    }
}
lo0 {
    unit 1 {
        family inet {
            address 172.16.0.1/32;
            address 192.168.1.1/32;
        }
    }
}

user@R1# show protocols
rip {
    traceoptions {
        file rip-trace-file;
        flag route;
    }
    group rip-group {
        export advertise-routes-through-rip;
        neighbor fe-1/2/0.1;
    }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Log File

Purpose Make sure that the RIP route updates are logged in the configured log file.

Action 1. Deactivate the extra loopback interface address on Device R3.

```

[edit interfaces lo0 unit 3 family inet]
user@R3# deactivate address 172.16.3.3/32
user@R3# commit

```

2. From operational mode on Device R1, enter the **show log rip-trace-file** command with the **| match 172.16.3.3** option.

```

user@R1> show log rip-trace-file | match 172.16.3.3
Mar  1 11:39:53.975192 Setting RIPv2 rtbit on route 172.16.3.3/32, tsi =
0xbb69228
Mar  1 11:39:59.847118 172.16.3.3/32: metric-in: 16, change: 3 -> 16; # gw:
1, pkt_upd_src 10.0.0.2, inx: 0, rte_upd_src 10.0.0.2
Mar  1 11:39:59.847568 CHANGE 172.16.3.3/32 nhid 591 gw 10.0.0.2

```

```
RIP      pref 100/0 metric 3/0 fe-1/2/0.1 <Delete Int>  
Mar  1 11:39:59.847629 Best route to 172.16.3.3/32 got deleted. Doing route calculation  
on the stored rte-info
```

Meaning The output shows that the route to 172.16.3.3/32 was deleted.

Related Documentation

- [Example: Configuring RIP on page 13](#)

Verifying a RIP Configuration

To verify a RIP configuration, perform the following tasks:

- [Verifying the Exchange of RIP Messages on page 94](#)
- [Verifying the RIP-Enabled Interfaces on page 96](#)
- [Verifying Reachability of All Hosts in the RIP Network on page 96](#)

Verifying the Exchange of RIP Messages

Purpose Verify that RIP messages are being sent and received on all RIP-enabled interfaces.

Action From the CLI, enter the **show rip statistics** command.

Sample Output

```

user@host> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped
           10           0           0           0

t1-0/0/2.0: 0 routes learned; 13 routes advertised; timeout 120s; update interval
45s
Counter                Total    Last 5 min  Last minute
-----
Updates Sent            2855         11         2
Triggered Updates Sent    5         0         0
Responses Sent           0         0         0
Bad Messages            0         0         0
RIPv1 Updates Received   0         0         0
RIPv1 Bad Route Entries  0         0         0
RIPv1 Updates Ignored    0         0         0
RIPv2 Updates Received   41         0         0
RIPv2 Bad Route Entries  0         0         0
RIPv2 Updates Ignored    0         0         0
Authentication Failures  0         0         0
RIP Requests Received    0         0         0
RIP Requests Ignored     0         0         0

ge-0/0/1.0: 10 routes learned; 3 routes advertised; timeout 180s; update interval
30s
Counter                Total    Last 5 min  Last minute
-----
Updates Sent            2855         11         2
Triggered Updates Sent    3         0         0
Responses Sent           0         0         0
Bad Messages            1         0         0
RIPv1 Updates Received   0         0         0
RIPv1 Bad Route Entries  0         0         0
RIPv1 Updates Ignored    0         0         0
RIPv2 Updates Received   2864        11         2
RIPv2 Bad Route Entries  14         0         0
RIPv2 Updates Ignored    0         0         0
Authentication Failures  0         0         0
RIP Requests Received    0         0         0
RIP Requests Ignored     0         0         0

```

Meaning The output shows the number of RIP routes learned. It also shows the number of RIP updates sent and received on the RIP-enabled interfaces. Verify the following information:

- The number of RIP routes learned matches the number of expected routes learned. Subnets learned by direct connectivity through an outgoing interface are not listed as RIP routes.
- RIP updates are being sent on each RIP-enabled interface. If no updates are being sent, the routing policy might not be configured to export routes.
- RIP updates are being received on each RIP-enabled interface. If no updates are being received, the routing policy might not be configured to export routes on the host connected to that subnet. The lack of updates might also indicate an authentication error.

Verifying the RIP-Enabled Interfaces

Purpose Verify that all the RIP-enabled interfaces are available and active.

Action From the CLI, enter the **show rip neighbor** command.

Sample Output

```
user@host> show rip neighbor
Source      Destination  Send  Receive  In
Neighbor    State  Address      Address      Mode  Mode  Met
-----
ge-0/0/0.0   Dn (null)   (null)      (null)      mcast both  1
ge-0/0/1.0   Up 192.168.220.5  224.0.0.9   mcast both  1
```

Meaning The output shows a list of the RIP neighbors that are configured on the device. Verify the following information:

- Each configured interface is present. Interfaces are listed in alphabetical order.
- Each configured interface is up. The state of the interface is listed in the **Destination State** column. A state of **Up** indicates that the link is passing RIP traffic. A state of **Dn** indicates that the link is not passing RIP traffic. In a point-to-point link, this state generally means that either the end point is not configured for RIP or the link is unavailable.

Verifying Reachability of All Hosts in the RIP Network

Purpose By using the traceroute tool on each loopback address in the network, verify that all hosts in the RIP network are reachable from each Juniper Networks device.

Action For each device in the RIP network:

1. In the J-Web interface, select **Troubleshoot>Traceroute**.
2. In the Remote Host box, type the name of a host for which you want to verify reachability from the device.
3. Click **Start**. Output appears on a separate page.

Sample Output

```
1 172.17.40.254 (172.17.40.254) 0.362 ms 0.284 ms 0.251 ms
2 routera-fxp0.englab.mycompany.net (192.168.71.246) 0.251 ms 0.235 ms 0.200 ms
```

Meaning Each numbered row in the output indicates a routing hop in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop for each traceroute packet.

To ensure that the RIP network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.

- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is probably unreachable. It might also indicate that the incoming or outgoing metric on one or more hosts has been set unexpectedly.

**Related
Documentation**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- [RIP Configuration Overview on page 8](#)
- [show rip statistics on page 156](#) in the Junos OS Operational Mode Commands
- [show rip neighbor on page 154](#) in the Junos OS Operational Mode Commands
- traceroute in the Junos OS Operational Mode Commands
- [RIP Overview on page 3](#)

CHAPTER 4

Configuration Statements

- [\[edit protocols rip\] Hierarchy Level on page 99](#)

[\[edit protocols rip\] Hierarchy Level](#)

The following statement hierarchy can also be included at the `[edit logical-systems logical-system-name]` hierarchy level.

```
protocols {
  rip {
    authentication-key password;
    authentication-type type;
    (check-zero | no-check-zero);
    graceful-restart {
      disable;
      restart-time seconds;
    }
    group group-name {
      ... the group subhierarchy appears after the main [edit protocols rip] hierarchy ...
    }
    holddown seconds;
    import [ policy-names ];
    message-size number;
    metric-in metric;
    receive (both | none | version-1 | version-2);
    rib-group group-name;
    route-timeout seconds;
    send (broadcast | multicast | none | version-1);
    traceoptions {
      file filename <files number> <size maximum-file-size> <world-readable |
        no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
    update-interval seconds;
  }

  rip {
    group group-name {
      bfd-liveness-detection {
        authentication {
          algorithm (keyed-md5 | keyed-sha-1 | meticulous-keyed-md5 |
            meticulous-keyed-sha-1 | simple-password);
```

```


        key-chain key-chain-name;
        loose-check;
    }
    detection-time {
        threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    transmit-interval {
        minimum-interval milliseconds;
        threshold milliseconds;
    }
    version (1 | automatic);
}
demand-circuit;
export [ policy-names ];
import [ policy-names ];
max-retrans-time seconds;
metric-out metric;
neighbor interface-name {
    ... the neighbor subhierarchy appears after the main [edit protocols rip group
        group-name] hierarchy level ...
}
preference preference;
route-timeout seconds;
update-interval seconds;
}

group group-name {
    neighbor neighbor-name {
        any-sender;
        authentication-key password;
        authentication-type type;
        bfd-liveness-detection {
            ... same statements as at the [edit protocols rip group group-name
                bfd-liveness-detection] hierarchy level ...
        }
        (check-zero | no-check-zero);
        demand-circuit;
        import [ policy-names ];
        max-retrans-time seconds;
        message-size number;
        metric-in metric;
        receive (both | none | version-1 | version-2);
        route-timeout seconds;
        send (broadcast | multicast | none | version-1);
        update-interval seconds;
    }
}
}
}

```

- Related Documentation**
- Notational Conventions Used in Junos OS Configuration Hierarchies
 - [edit protocols] Hierarchy Level

any-sender

Syntax	any-sender;
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 8.0.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Disable strict sender address checks.</p> <p>If the sender of a RIP message does not belong to the subnet of the interface, the message is discarded. This situation might cause problems with dropped packets when RIP is running on point-to-point interfaces, or when the addresses on the interfaces do not fall in the same subnet. You can resolve this by disabling strict address checks on the RIP traffic.</p>
	<div>  <p>NOTE: The any-sender statement is supported only for peer-to-peer interfaces.</p> </div>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 13

authentication-key (Protocols RIP)

Syntax	<code>authentication-key password;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Require authentication for RIP route queries received on an interface.
Options	<i>password</i> —Authentication password. If the password does not match, the packet is rejected. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.
Required Privilege Level	<code>routing</code> —To view this statement in the configuration. <code>routing-control</code> —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring Route Authentication for RIP on page 20

authentication-type (Protocols RIP)

Syntax	<code>authentication-type type;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure the type of authentication for RIP route queries received on an interface.
Default	If you do not include this statement and the authentication-key statement, RIP authentication is disabled.
Options	<p>type—Authentication type:</p> <ul style="list-style-type: none"> • md5—Use the MD5 algorithm to create an encoded checksum of the packet. The encoded checksum is included in the transmitted packet. The receiving routing device uses the authentication key to verify the packet, discarding it if the digest does not match. This algorithm provides a more secure authentication scheme. • none—Disable authentication. If none is configured, the configured authentication key is ignored. • simple—Use a simple password. The password is included in the transmitted packet, which makes this method of authentication relatively insecure. The password can be from 1 through 16 contiguous letters or digits long.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring Route Authentication for RIP on page 20 • authentication-key on page 102

bfd-liveness-detection (Protocols RIP)

Syntax	<pre> bfd-liveness-detection { authentication { algorithm <i>algorithm-name</i>; key-chain <i>key-chain-name</i>; loose-check; } detection-time { threshold <i>milliseconds</i>; } minimum-interval <i>milliseconds</i>; minimum-receive-interval <i>milliseconds</i>; multiplier <i>number</i>; no-adaptation; transmit-interval { minimum-interval <i>milliseconds</i>; threshold <i>milliseconds</i>; } version (1 automatic); } </pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>], [edit protocols rip group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 8.0.</p> <p>Options detection-time threshold and transmit-interval threshold introduced in Junos OS Release 8.2.</p> <p>Support for logical systems introduced in Junos OS Release 8.3.</p> <p>Option no-adaptation introduced in Junos OS Release 9.0.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Options authentication algorithm, authentication key-chain, and authentication loose-check introduced in Junos OS Release 9.6.</p> <p>Options authentication algorithm, authentication key-chain, and authentication loose-check introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure bidirectional failure detection timers and authentication.
Options	<p>authentication algorithm <i>algorithm-name</i> —Configure the algorithm used to authenticate the specified BFD session: simple-password, keyed-md5, keyed-sha-1, meticulous-keyed-md5, or meticulous-keyed-sha-1.</p> <p>authentication key-chain <i>key-chain-name</i>—Associate a security key with the specified BFD session using the name of the security keychain. The name you specify must match one of the keychains configured in the authentication-key-chains key-chain statement at the [edit security] hierarchy level.</p>

authentication loose-check—(Optional) Configure loose authentication checking on the BFD session. Use only for transitional periods when authentication is not configured at both ends of the BFD session.

detection-time threshold *milliseconds*—Configure a threshold for the adaptation of the BFD session detection time. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

minimum-interval *milliseconds*—Configure the minimum interval after which the local routing device transmits a hello packet and then expects to receive a reply from the neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and **minimum-receive-interval** statements.

Range: 1 through 255,000 milliseconds

minimum-receive-interval *milliseconds*—Configure the minimum interval after which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can configure the minimum receive interval using the **minimum-interval** statement.

Range: 1 through 255,000 milliseconds

multiplier *number*—Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

Range: 1 through 255

Default: 3

no-adaptation—Configure BFD sessions not to adapt to changing network conditions. We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

transmit-interval threshold *milliseconds*—Configure the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent. The interval threshold must be greater than the minimum transmit interval.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

transmit-interval minimum-interval *milliseconds*—Configure a minimum interval after which the local routing device transmits hello packets to a neighbor. Optionally, instead of using this statement, you can configure the minimum transmit interval using the **minimum-interval** statement.

Range: 1 through 255,000

version—Configure the BFD version to detect: **1** (BFD version 1) or **automatic** (autodetect the BFD version).

Default: automatic

Required Privilege Level	routing—To view this statement in the configuration.
	routing-control—To add this statement to the configuration.

- Related Documentation**
- [Example: Configuring BFD for RIP on page 27](#)
 - [Example: Configuring BFD Authentication for RIP on page 34](#)

check-zero

Syntax	(check-zero no-check-zero);
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	<p>Some of the reserved fields in RIP version 1 packets must be zero, whereas in RIP version 2 packets, most of these reserved fields can contain nonzero values. By default, RIP discards version 1 packets that have nonzero values in the reserved fields and version 2 packets that have nonzero values in the fields that must be zero. This default behavior implements the RIP version 1 and version 2 specifications.</p> <p>If you find that you are receiving RIP version 1 packets with nonzero values in the reserved fields or RIP version 2 packets with nonzero values in the fields that must be zero, you can configure RIP to receive these packets even though they are being sent in violation of the specifications in RFC 1058 and RFC 2453.</p> <p>Check whether the reserved fields in a RIP packet are zero:</p> <ul style="list-style-type: none"> • check-zero—Discard version 1 packets that have nonzero values in the reserved fields and version 2 packets that have nonzero values in the fields that must be zero. This default behavior implements the RIP version 1 and version 2 specifications. • no-check-zero—Receive RIP version 1 packets with nonzero values in the reserved fields or RIP version 2 packets with nonzero values in the fields that must be zero. This is in spite of the fact that they are being sent in violation of the specifications in RFC 1058 and RFC 2453.
Default	check-zero
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 13


demand-circuit (Protocols RIP)

Syntax	<code>demand-circuit;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip group <i>group-name</i>],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Configure a neighboring interface to act as a RIP demand circuit. To complete the demand circuit, you must configure both ends of the pair as demand circuits. When configured, the device sends RIP information only when changes occur in the routing database.
Default	Disabled. You must explicitly configure two neighboring interfaces to act as a RIP demand circuit.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP Demand Circuits on page 79• RIP Demand Circuits Overview on page 77• max-retrans-time on page 116

dynamic-peers

Syntax	<code>dynamic-peers;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 11.4.
Description	Configure an interface to have dynamic peers in a point-to-multipoint RIP network.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring Point-to-Multipoint RIP Networks on page 41

export (Protocols RIP)

Syntax	<code>export [<i>policy-names</i>];</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit protocols rip group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Apply a policy to routes being exported to the neighbors.</p> <p>By default, RIP does not export routes it has learned to its neighbors. To enable RIP to export routes, apply one or more export policies.</p> <p>If no routes match the policies, the local routing device does not export any routes to its neighbors. Export policies override any metric values determined through calculations involving the values configured with the metric-in and metric-out statements.</p> <div style="margin-top: 20px;">  <p>NOTE: The export policy on RIP does not support manipulating routing information of the next hop.</p> </div>
Options	<i>policy-names</i> —Name of one or more policies.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 13 • import on page 114

graceful-restart (Protocols RIP)

Syntax	<pre>graceful-restart { disable; restart-time <i>seconds</i>; }</pre>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip], [edit protocols rip]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	Configure graceful restart for RIP.
Options	disable —Disables graceful restart for RIP. The remaining statement is explained separately.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Junos OS High Availability Configuration Guide

group (Protocols RIP)

```
Syntax  group group-name {
        bfd-liveness-detection {
            authentication {
                algorithm algorithm-name;
                key-chain key-chain-name;
                loose-check;
            }
            detection-time {
                threshold milliseconds;
            }
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            transmit-interval {
                threshold milliseconds;
                minimum-interval milliseconds;
            }
            multiplier number;
            version (0 | 1 | automatic);
        }
        demand-circuit;
        export policy;
        max-retrans-time seconds;
        metric-out metric;
        preference number;
        route-timeout seconds;
        update-interval seconds;
        neighbor neighbor-name {
            authentication-key password;
            authentication-type type;
            bfd-liveness-detection {
                authentication {
                    algorithm algorithm-name;
                    key-chain key-chain-name;
                    loose-check;
                }
                detection-time {
                    threshold milliseconds;
                }
                minimum-interval milliseconds;
                minimum-receive-interval milliseconds;
                transmit-interval {
                    threshold milliseconds;
                    minimum-interval milliseconds;
                }
                multiplier number;
                version (0 | 1 | automatic);
            }
            (check-zero | no-check-zero);
            demand-circuit;
            import policy-name;
            max-retrans-time seconds;
            message-size number;
        }
    }
```

```
metric-in metric;  
metric-out metric;  
receive receive-options;  
route-timeout seconds;  
send send-options;  
update-interval seconds;  
}  
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols [rip](#)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols
[rip](#)],
[edit protocols [rip](#)],
[edit routing-instances *routing-instance-name* protocols [rip](#)]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure a set of RIP neighbors that share an export policy and metric. The export policy and metric govern what routes to advertise to neighbors in a given group. Each group must contain at least one neighbor. You should create a group for every export policy.

Options *group-name*—Name of a group, up to 16 characters long.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [Example: Configuring RIP on page 13](#)

holddown (Protocols RIP)

Syntax	<code>holddown seconds;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	<p>Configure how long the expired route is retained in the routing table before being removed.</p> <p>When the hold-down timer runs on RIP demand circuits, routes are advertised as unreachable on other interfaces. When the hold-down timer expires, the route is removed from the routing table if all destinations detect that the route is unreachable or the remaining destinations are down.</p>
Options	<p>seconds—Estimated time to wait before making updates to the routing table.</p> <p>Range: 10 through 180 seconds</p> <p>Default: 180 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP Timers on page 71 • RIP Demand Circuits Overview on page 77

import (Protocols RIP)

Syntax	<code>import [<i>policy-names</i>];</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Apply one or more policies to routes being imported by the local routing device from neighbors.
Options	<i>policy-names</i> —Name of one or more policies.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Applying Policies to RIP Routes Imported from Neighbors on page 47• Routing Policy Configuration Guide• export on page 109


interface-type (Protocols RIP)

Syntax	interface-type p2mp;
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>], [edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]
Release Information	Statement introduced in Junos OS Release 11.4.
Description	Configure the type of interface in a RIP network. This statement enables a RIP device to have single or multiple peers through an interface.
Options	p2mp —Configure an interface in a RIP network as a point-to-multipoint interface.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring Point-to-Multipoint RIP Networks on page 41

max-retrans-time

Syntax	<code>max-retrans-time seconds;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip group <i>group-name</i>],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 11.1.
Description	<p>RIP demand circuits send update messages every 5 seconds to an unresponsive peer. Configure the retransmission timer to limit the number of times the demand circuit resends update messages to an unresponsive peer. If the configured retransmission threshold is reached, routes from the next hop router are marked as unreachable and the hold-down timer starts. You must configure a pair of RIP demand circuits for this timer to take effect.</p> <p>To determine the number of times to resend the update message, use the following calculation:</p> $5 \text{ seconds} \times \text{number of retransmissions} = \text{retransmission seconds}$
Options	<p>seconds—The total amount of time the demand circuit resends update messages to an unresponsive peer. The seconds range corresponds to sending an update message a minimum of 1 time (5 seconds) and a maximum of 36 times (180 seconds).</p> <p>Range: 5 through 180 seconds</p> <p>Default: 5 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP Demand Circuits on page 79• RIP Demand Circuits Overview on page 77• demand-circuit on page 108

message-size

Syntax	<code>message-size <i>number</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement for SRX Series devices introduced in Junos OS Release 9.5.</p> <p>Statement for J Series platform introduced in Junos OS Release 8.5.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Specify the number of route entries to be included in every RIP update message.
	<div>  <p>TIP: To ensure interoperability with other vendors' equipment, use the standard of 25 route entries per message. Do not change the default number of route entries in a RIP update message.</p> </div>
Options	<p><i>number</i>—Number of route entries per update message.</p> <p>Range: 25 through 255 entries</p> <p>Default: 25 entries</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Example: Configuring RIP on page 13

metric-in (Protocols RIP)

Syntax	<code>metric-in <i>metric</i>;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code> rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code> <i>neighbor-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Specify the metric to add to incoming routes when the routing device advertises into RIP routes that were learned from other protocols. Use this statement to configure the routing device to prefer RIP routes learned through a specific neighbor.
Options	<i>metric</i> —Metric value. Range: 1 through 16 Default: 1
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring the Metric Value Added to Imported RIP Routes on page 56

metric-out (Protocols RIP)

Syntax	<code>metric-out <i>metric</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Specify the metric value to add to routes transmitted to the neighbor. Use this statement to control how other routing devices prefer RIP routes sent from this neighbor.</p> <p>If you have included the export statement, RIP exports routes it has learned to the neighbors configured by including the neighbor statement.</p> <p>The metric associated with a RIP route (unless modified by an export policy) is the normal RIP metric. For example, a RIP route with a metric of 5 learned from a neighbor configured with a metric-in value of 2 is advertised with a combined metric of 7 when advertised to RIP neighbors in the same group. However, if this route was learned from a RIP neighbor in a different group or from a different protocol, the route is advertised with the metric value configured for that group with the metric-out statement.</p> <p>The metric for a route can be modified with an export policy. That metric is seen when the route is exported to the next hop.</p> <p>To increase the metric for routes advertised outside a group, include the metric-out statement.</p>
Options	<p><i>metric</i>—Metric value.</p> <p>Range: 1 through 16</p> <p>Default: 1</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Examples: Controlling Traffic with Metrics in a RIP Network on page 5250

neighbor (Protocols RIP)

Syntax `neighbor neighbor-name {`
 `authentication-key password;`
 `authentication-type type;`
 `bfd-liveness-detection {`
 `authentication {`
 `algorithm algorithm-name;`
 `key-chain key-chain-name;`
 `loose-check;`
 `}`
 `detection-time {`
 `threshold milliseconds;`
 `}`
 `minimum-interval milliseconds;`
 `minimum-receive-interval milliseconds;`
 `transmit-interval {`
 `threshold milliseconds;`
 `minimum-interval milliseconds;`
 `}`
 `multiplier number;`
 `version (0 | 1 | automatic);`
 `}`
 `(check-zero | no-check-zero);`
 `demand-circuit;`
 `import policy-name;`
 `max-retrans-time seconds;`
 `message-size number;`
 `metric-in metric;`
 `metric-out metric;`
 `receive receive-options;`
 `route-timeout seconds;`
 `send send-options;`
 `update-interval seconds;`
 `}`

Hierarchy Level [edit logical-systems *logical-system-name* protocols rip **group** *group-name*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols
 rip **group** *group-name*],
 [edit protocols rip **group** *group-name*],
 [edit routing-instances *routing-instance-name* protocols rip **group** *group-name*]

Release Information Statement introduced before Junos OS Release 7.4.
 Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Configure neighbor-specific RIP parameters, thereby overriding the defaults set for the routing device.

Options *neighbor-name*—Name of an interface over which a routing device communicates to its neighbors.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [Example: Configuring RIP on page 13](#)

peer (Protocols RIP)

Syntax *peer IP address;*

Hierarchy Level [edit logical-systems *logical-system-name* protocols rip group *group-name* neighbor *neighbor-name*],
[edit protocols rip group *group-name* neighbor *neighbor-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Configure a static peer for an interface in a point-to-multipoint RIP network.

Options *address*—IP address of the static peer to be configured.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- [Example: Configuring Point-to-Multipoint RIP Networks on page 41](#)

preference (Protocols RIP)

Syntax	<code>preference <i>preference</i>;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>], [edit protocols rip group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Specify the preference of external routes learned by RIP as compared to those learned from other routing protocols.</p> <p>By default, Junos OS assigns a preference of 100 to routes that originate from RIP. When Junos OS determines a route's preference to become the active route, the software selects the route with the lowest preference and installs this route into the forwarding table.</p>
Options	<p>preference—Preference value. A lower value indicates a more preferred route.</p> <p>Range: 0 through 4,294,967,295 ($2^{32} - 1$)</p> <p>Default: 100</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Route Preferences Overview

receive (Protocols RIP)

Syntax	<code>receive receive-options;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure RIP receive options.
Options	<p><i>receive-options</i>—One of the following:</p> <ul style="list-style-type: none"> • both—Accept both RIP version 1 and version 2 packets. • none—Do not receive RIP packets. • version-1—Accept only RIP version 1 packets. • version-2—Accept only RIP version 2 packets. <p>Default: both</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 61 • send on page 127

rib-group (Protocols RIP)

Syntax	<code>rib-group group-name;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip], [edit protocols rip], [edit routing-instances <i>routing-instance-name</i> protocols rip]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Install RIP routes into multiple routing tables by configuring a routing table group.
Options	<i>group-name</i> —Name of the routing table group.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Redistributing Routes Between Two RIP Instances on page 65

rip

Syntax	<code>rip {...}</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols], [edit protocols], [edit routing-instances <i>routing-instance-name</i> protocols]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Enable RIP routing on the routing device.
Default	RIP is disabled on the routing device.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP on page 13

route-timeout (Protocols RIP)

Syntax	<code>route-timeout seconds;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group group-name],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group group-name neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group group-name],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group group-name neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group group-name],</p> <p>[edit protocols rip group group-name neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group group-name],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group group-name neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure the route timeout interval for RIP. If a route is not refreshed after being installed in the routing table by the specified timeout interval, the route is marked as invalid and is removed from the routing table after the hold-down period expires.
Options	<p>seconds—Estimated time to wait before making updates to the routing table.</p> <p>Range: 30 through 360 seconds</p> <p>Default: 180 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP Timers on page 71 • RIP Demand Circuits Overview on page 77

routing-instances (Multiple Routing Entities)

Syntax	<code>routing-instances <i>routing-instance-name</i> { ... }</code>
Hierarchy Level	[edit], [edit logical-systems <i>logical-system-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	<p>Configure an additional routing entity for a router. You can create multiple instances of BGP, IS-IS, OSPF, OSPFv3, and RIP for a router. You can also create multiple routing instances for separating routing tables, routing policies, and interfaces for individual wholesale subscribers (retailers) in a Layer 3 wholesale network.</p> <p>Each routing instance consist of the following:</p> <ul style="list-style-type: none">• A set of routing tables• A set of interfaces that belong to these routing tables• A set of routing option configurations <p>Each routing instance has a unique name and a corresponding IP unicast table. For example, if you configure a routing instance with the name my-instance, its corresponding IP unicast table is my-instance.inet.0. All routes for my-instance are installed into my-instance.inet.0.</p> <p>Routes are installed into the default routing instance inet.0 by default, unless a routing instance is specified.</p> <p>In Junos OS Release 9.0 and later, you can no longer specify a routing-instance name of <i>master</i>, <i>default</i>, or <i>bgp</i> or include special characters within the name of a routing instance.</p> <p>In Junos OS Release 9.6 and later, you can include a slash (/) in a routing-instance name only if a logical system is not configured. That is, you cannot include the slash character in a routing-instance name if a logical system other than the default is explicitly configured. Routing-instance names, further, are restricted from having the form <code>__.*__</code> (beginning and ending with underscores). The colon : character cannot be used when multipotology routing (MTR) is enabled.</p>
Default	Routing instances are disabled for the router.
Options	<p><i>routing-instance-name</i>—Name of the routing instance. This must be a non-reserved string of not more than 128 characters.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- | | |
|------------------------------|--|
| Related Documentation | <ul style="list-style-type: none"> • Example: Configuring Interprovider Layer 3 VPN Option A • Example: Configuring Interprovider Layer 3 VPN Option B • Example: Configuring Interprovider Layer 3 VPN Option C • Example: Configuring E-LINE and E-LAN Services for a PBB Network on MX Series Routers |
|------------------------------|--|

send (Protocols RIP)

Syntax	<code>send <i>send-options</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure RIP send options.
Options	<p><i>send-options</i>—One of the following:</p> <ul style="list-style-type: none"> • broadcast—Broadcast RIP version 2 packets (RIP version 1 compatible). • multicast—Multicast RIP version 2 packets. This is the default. • none—Do not send RIP updates. • version-1—Broadcast RIP version 1 packets. <p>Default: multicast</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 61 • receive on page 123

traceoptions (Protocols RIP)

Syntax	<pre>traceoptions { file <i>filename</i> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i> <flag-modifier> <disable>; }</pre>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip], [edit protocols rip], [edit routing-instances <i>routing-instance-name</i> protocols rip]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series.
Description	Set RIP protocol-level tracing options.



NOTE: The **traceoptions** statement is not supported on QFabric systems.

Default The default RIP protocol-level trace options are inherited from the global **traceoptions** statement.

Options **disable**—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place RIP tracing output in the file `/var/log/rip-log`.

files *number*—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 10 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

RIP Tracing Options

- **auth**—RIP authentication
- **error**—RIP error packets

- **expiration**—RIP route expiration processing
- **holddown**—RIP hold-down processing
- **nsr-synchronization**—Nonstop routing synchronization events
- **packets**—All RIP packets
- **request**—RIP information packets such as request, poll, and poll entry packets
- **trigger**—RIP triggered updates
- **update**—RIP update packets

Global Tracing Options

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Provide detailed trace information.
- **receive**—Trace the packets being received.
- **receive-detail**—Provide detailed trace information for packets being received.
- **send**—Trace the packets being transmitted.
- **send-detail**—Provide detailed trace information for packets being transmitted.

no-world-readable—(Optional) Prevent any user from reading the log file.

size size—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named **trace-file** reaches this size, it is renamed **trace-file.0**. When the **trace-file** again reaches its maximum size, **trace-file.0** is renamed **trace-file.1** and **trace-file** is renamed **trace-file.0**. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: **xk** to specify KB, **xm** to specify MB, or **xg** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 128 KB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level	routing—To view this statement in the configuration.
	routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Tracing RIP Protocol Traffic on page 90

update-interval (Protocols RIP)

Syntax	<code>update-interval seconds;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i>],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
Description	Configure the interval at which routes learned by RIP are sent to neighbors. This timer controls the interval between routing updates. This timer is set to 30 seconds, by default, with a small random amount of time added when the timer is reset. This added time prevents congestion that can happen if all routing devices update their neighbors simultaneously.
Options	<p>seconds—Estimated time to wait before making updates to the routing table.</p> <p>Range: 10 through 60 seconds</p> <p>Default: 30 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP Timers on page 71

PART 3

Administration

- [Operational Commands on page 135](#)

CHAPTER 5

Operational Commands

clear rip general-statistics

Syntax	clear rip general-statistics <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches and QFX Series)	clear rip general-statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series.
Description	Clear RIP general statistics.
Options	none —Clear RIP general statistics. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show rip general-statistics on page 152
List of Sample Output	clear rip general-statistics on page 136
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear rip general-statistics	user@host> clear rip general-statistics
---------------------------------	---

clear rip statistics

Syntax	clear rip statistics <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> <neighbor> <peer (all <i>address</i>)>
Syntax (EX Series Switches and QFX Series)	clear rip statistics <instance (all <i>instance-name</i>)> <neighbor>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series.
Description	Clear RIP statistics.
Options	<p>none—Reset RIP counters for all neighbors for all routing instances.</p> <p>instance (all <i>instance-name</i>)—(Optional) Clear RIP statistics for all instances or for the specified routing instance only.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p>neighbor—(Optional) Clear RIP statistics for the specified neighbor only.</p> <p>peer (all <i>address</i>)—(Optional) Clear RIP statistics for a single peer or all peers.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • show rip statistics on page 156
List of Sample Output	clear rip statistics on page 137
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear rip statistics user@host> clear rip statistics

restart

Syntax	<pre>restart <adaptive-services ancpd-service application-identification audit-process auto-configuration captive-portal-content-delivery ce-l2tp-service chassis-control class-of-service clksyncd-service database-replication datapath-trace-service dhcp-service diameter-service disk-monitoring dynamic-flow-capture ecc-error-logging ethernet-connectivity-fault-management ethernet-link-fault-management event-processing firewall general-authentication-service gracefully iccp-service idp-policy immediately interface-control ipsec-key-management kernel-replication l2-learning l2cpd-service l2tp-service l2tp-universal-edge lacp license-service link-management local-policy-decision-function mac-validation mib-process mobile-ip mountd-service mpls-traceroute mspd multicast-snooping named-service nfsd-service packet-triggered-subscribers peer-selection-service pgcp-service pgm pic-services-logging pki-service ppp ppp-service pppoe protected-system-domain-service redundancy-interface-process remote-operations root-system-domain-service routing <logical-system <i>logical-system-name</i>> sampling sbc-configuration-process sdk-service service-deployment services services pgcp gateway <i>gateway-name</i> snmp soft static-subscribers statistics-service subscriber-management subscriber-management-helper tunnel-oamd usb-control vrrp web-management> <gracefully immediately soft></pre>
Syntax (ACX Series Routers)	<pre>restart <adaptive-services audit-process auto-configuration autoinstallation chassis-control class-of-service clksyncd-service database-replication dhcp-service diameter-service disk-monitoring dynamic-flow-capture ethernet-connectivity-fault-management ethernet-link-fault-management event-processing firewall general-authentication-service gracefully immediately interface-control ipsec-key-management l2-learning lacp link-management mib-process mobile-ip mountd-service mpls-traceroute mspd named-service nfsd-service pgm pki-service ppp pppoe redundancy-interface-process remote-operations routing sampling sdk-service secure-neighbor-discovery service-deployment services snmp soft statistics-service subscriber-management subscriber-management-helper tunnel-oamd vrrp></pre>
Syntax (EX Series Switches)	<pre>restart <autoinstallation chassis-control class-of-service database-replication dhcp dhcp-service diameter-service dot1x-protocol ethernet-link-fault-management ethernet-switching event-processing firewall general-authentication-service interface-control kernel-replication l2-learning lacp license-service link-management lldpd-service mib-process mountd-service multicast-snooping pgm redundancy-interface-process remote-operations routing secure-neighbor-discovery service-deployment sflow-service snmp vrrp web-management></pre>
Syntax (Routing Matrix)	<pre>restart <adaptive-services audit-process chassis-control class-of-service disk-monitoring dynamic-flow-capture ecc-error-logging event-processing firewall interface-control ipsec-key-management kernel-replication l2-learning l2tp-service lacp link-management mib-process pgm pic-services-logging ppp pppoe redundancy-interface-process remote-operations routing <logical-system <i>logical-system-name</i>> sampling service-deployment snmp> <all all-lcc lcc <i>number</i>></pre>

	<gracefully immediately soft>
Syntax (J Series Routing Platform)	<p>restart</p> <p><adaptive-services audit-process chassis-control class-of-service dhcp dialer-services dlsw event-processing firewall interface-control ipsec-key-management isdn-signaling l2-learning l2tp-service mib-process network-access-service pgm ppp pppoe remote-operations routing <logical-system <i>logical-system-name</i>> sampling service-deployment snmp usb-control web-management></p> <p><gracefully immediately soft></p>
Syntax (TX Matrix Routers)	<p>restart</p> <p><adaptive-services audit-process chassis-control class-of-service dhcp-service diameter-service disk-monitoring dynamic-flow-capture ecc-error-logging event-processing firewall interface-control ipsec-key-management kernel-replication l2-learning l2tp-service lacp link-management mib-process pgm pic-services-logging ppp pppoe redundancy-interface-process remote-operations routing <logical-system <i>logical-system-name</i>> sampling service-deployment snmp statistics-service></p> <p><all-chassis all-lcc lcc <i>number</i> scc></p> <p><gracefully immediately soft></p>
Syntax (TX Matrix Plus Routers)	<p>restart</p> <p><adaptive-services audit-process chassis-control class-of-service dhcp-service diameter-service disk-monitoring dynamic-flow-capture ecc-error-logging event-processing firewall interface-control ipsec-key-management kernel-replication l2-learning l2tp-service lacp link-management mib-process pgm pic-services-logging ppp pppoe redundancy-interface-process remote-operations routing <logical-system <i>logical-system-name</i>> sampling service-deployment snmp statistics-service></p> <p><all-chassis all-lcc all-sfc lcc <i>number</i> sfc <i>number</i>></p> <p><gracefully immediately soft></p>
Syntax (MX Series Routers)	<p>restart</p> <p><adaptive-services ancpd-service application-identification audit-process auto-configuration captive-portal-content-delivery ce-l2tp-service chassis-control class-of-service clksyncd-service database-replication datapath-trace-service dhcp-service diameter-service disk-monitoring dynamic-flow-capture ecc-error-logging ethernet-connectivity-fault-management ethernet-link-fault-management event-processing firewall general-authentication-service gracefully iccp-service idp-policy immediately interface-control ipsec-key-management kernel-replication l2-learning l2cpd-service l2tp-service l2tp-universal-edge lacp license-service link-management local-policy-decision-function mac-validation mib-process mobile-ip mounstd-service mpls-traceroute mspd multicast-snooping named-service nfsd-service packet-triggered-subscribers peer-selection-service pgcp-service pgm pic-services-logging pki-service ppp ppp-service pppoe protected-system-domain-service redundancy-interface-process remote-operations root-system-domain-service routing routing <logical-system <i>logical-system-name</i>> sampling sbc-configuration-process sdk-service service-deployment services services pgcp gateway <i>gateway-name</i> snmp soft static-subscribers statistics-service subscriber-management subscriber-management-helper tunnel-oamd usb-control vrrp web-management></p> <p><all-members></p> <p><gracefully immediately soft></p> <p><local></p> <p><member <i>member-id</i>></p>

Syntax (J Series Routers)	<p>restart</p> <p><adaptive-services audit-process chassis-control class-of-service dhcp dhcp-service dialer-services diameter-service dls w event-processing firewall interface-control ipsec-key-management isdn-signaling l2ald l2-learning l2tp-service mib-process network-access-service pgm ppp pppoe remote-operations routing <logical-system <i>logical-system-name</i>> sampling service-deployment snmp usb-control web-management></p> <p><gracefully immediately soft></p>
Syntax (QFX Series)	<p>restart</p> <p><adaptive-services audit-process chassis-control class-of-service dialer-services diameter-service dls w ethernet-connectivity event-processing fibre-channel firewall general-authentication-service igmp-host-services interface-control ipsec-key-management isdn-signaling l2ald l2-learning l2tp-service mib-process named-service network-access-service nstrace-process pgm ppp pppoe redundancy-interface-process remote-operations <logical-system-name> routing sampling secure-neighbor-discovery service-deployment snmp usb-control web-management></p> <p><gracefully immediately soft></p>
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Command introduced in Junos OS Release 11.1 for the QFX Series.</p> <p>Command introduced in Junos OS Release 12.2 for ACX Series routers.</p> <p>Options added:</p> <ul style="list-style-type: none"> • dynamic-flow-capture in Junos OS Release 7.4. • dls w in Junos OS Release 7.5. • event-processing in Junos OS Release 7.5. • ppp in Junos OS Release 7.5. • l2ald in Junos OS Release 8.0. • link-management in Release 8.0. • pgcp-service in Junos OS Release 8.4. • sbc-configuration-process in Junos OS Release 9.5. • services pgcp gateway in Junos OS Release 9.6. • sfc and all-sfc for the TX Matrix Router in Junos OS Release 9.6.
Description	Restart a Junos OS process.
	
<p>CAUTION: Never restart a software process unless instructed to do so by a customer support engineer. A restart might cause the router or switch to drop calls and interrupt transmission, resulting in possible loss of data.</p>	
Options	none—Same as gracefully .

- adaptive-services**—(Optional) Restart the configuration management process that manages the configuration for stateful firewall, Network Address Translation (NAT), intrusion detection services (IDS), and IP Security (IPsec) services on the Adaptive Services PIC.
- all-chassis**—(TX Matrix and TX Matrix Plus routers only) (Optional) Restart the software process on all chassis.
- all-lcc**—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process on all T640 routers connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process on all T1600 routers connected to the TX Matrix Plus router.
- all-members**—(MX Series routers only) (Optional) Restart the software process for all members of the Virtual Chassis configuration.
- all-sfc**—(TX Matrix Plus routers only) (Optional) For a TX Matrix Plus router, restart the software processes for the TX Matrix Plus router (or switch-fabric chassis).
- ancpd-service**—(Optional) Restart the Access Node Control Protocol (ANCP) process, which works with a special Internet Group Management Protocol (IGMP) session to collect outgoing interface mapping events in a scalable manner.
- application-identification**—(Optional) Restart the process that identifies an application using intrusion detection and prevention (IDP) to allow or deny traffic based on applications running on standard or nonstandard ports.
- audit-process**—(Optional) Restart the RADIUS accounting process that gathers statistical data that can be used for general network monitoring, analyzing, and tracking usage patterns, for billing a user based on the amount of time or type of services accessed.
- auto-configuration**—(Optional) Restart the Interface Auto-Configuration process.
- autoinstallation**—(EX Series switches only) (Optional) Restart the autoinstallation process.
- captive-portal-content-delivery**—(Optional) Restart the HTTP redirect service by specifying the location to which a subscriber's initial Web browser session is redirected, enabling initial provisioning and service selection for the subscriber.
- ce-l2tp-service**—(M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Universal Edge Layer 2 Tunneling Protocol (L2TP) process, which establishes L2TP tunnels and Point-to-Point Protocol (PPP) sessions through L2TP tunnels.
- chassis-control**—(Optional) Restart the chassis management process.
- class-of-service**—(Optional) Restart the class-of-service (CoS) process, which controls the router's or switch's CoS configuration.
- clksyncd-service**—(Optional) Restart the external clock synchronization process, which uses synchronous Ethernet (SyncE).

database-replication—(EX Series switches and MX Series routers only) (Optional) Restart the database replication process.

datapath-trace-service—(Optional) Restart the packet path tracing process.

dhcp—(J Series routers and EX Series switches only) (Optional) Restart the software process for a Dynamic Host Configuration Protocol (DHCP) server. A DHCP server allocates network IP addresses and delivers configuration settings to client hosts without user intervention.

dhcp-service—(Optional) Restart the Dynamic Host Configuration Protocol process.

dialer-services—(J Series routers and EX Series switches only) (Optional) Restart the ISDN dial-out process.

diameter-service—(Optional) Restart the diameter process.

disk-monitoring—(Optional) Restart disk monitoring, which checks the health of the hard disk drive on the Routing Engine.

dls—(J Series routers and QFX Series only) (Optional) Restart the data link switching (DLSw) service.

dot1x-protocol—(EX Series switches only) (Optional) Restart the port-based network access control process.

dynamic-flow-capture—(Optional) Restart the dynamic flow capture (DFC) process, which controls DFC configurations on Monitoring Services III PICs.

ecc-error-logging—(Optional) Restart the error checking and correction (ECC) process, which logs ECC parity errors in memory on the Routing Engine.

ethernet-connectivity-fault-management—(Optional) Restart the process that provides IEEE 802.1ag Operation, Administration, and Management (OAM) connectivity fault management (CFM) database information for CFM maintenance association end points (MEPs) in a CFM session.

ethernet-link-fault-management—(EX Series switches and MX Series routers only) (Optional) Restart the process that provides the OAM link fault management (LFM) information for Ethernet interfaces.

ethernet-switching—(EX Series switches only) (Optional) Restart the Ethernet switching process.

event-processing—(Optional) Restart the event process (eventd).

fibre-channel—(QFX Series only) (Optional) Restart the Fibre Channel process.

firewall—(Optional) Restart the firewall management process, which manages the firewall configuration and enables accepting or rejecting packets that are transiting an interface on a router or switch.

general-authentication-service—(EX Series switches and MX Series routers only) (Optional) Restart the general authentication process.

gracefully—(Optional) Restart the software process.

iccp-service—(Optional) Restart the Inter-Chassis Communication Protocol (ICCP) process.

idp-policy—(Optional) Restart the intrusion detection and prevention (IDP) protocol process.

immediately—(Optional) Immediately restart the software process.

interface-control—(Optional) Restart the interface process, which controls the router's or switch's physical interface devices and logical interfaces.

ipsec-key-management—(Optional) Restart the IPsec key management process.

isdn-signaling—(J Series routers and QFX Series only) (Optional) Restart the ISDN signaling process, which initiates ISDN connections.

kernel-replication—(Optional) Restart the kernel replication process, which replicates the state of the backup Routing Engine when graceful Routing Engine switchover (GRES) is configured.

l2-learning—(Optional) Restart the Layer 2 address flooding and learning process.

l2cpd-service—(Optional) Restart the Layer 2 Control Protocol process, which enables features such as Layer 2 protocol tunneling and nonstop bridging.

l2tp-service—(M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Layer 2 Tunneling Protocol (L2TP) process, which sets up client services for establishing Point-to-Point Protocol (PPP) tunnels across a network and negotiating Multilink PPP if it is implemented.

l2tp-universal-edge—(MX Series routers only) (Optional) Restart the L2TP process, which establishes L2TP tunnels and PPP sessions through L2TP tunnels.

lACP—(Optional) Restart the Link Aggregation Control Protocol (LACP) process. LACP provides a standardized means for exchanging information between partner systems on a link to allow their link aggregation control instances to reach agreement on the identity of the LAG to which the link belongs, and then to move the link to that LAG, and to enable the transmission and reception processes for the link to function in an orderly manner.

lcc number—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process for a specific T640 router that is connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process for a specific T1600 router that is connected to the TX Matrix Plus router. Replace **number** with a value from 0 through 3.

license-service—(EX Series switches only) (Optional) Restart the feature license management process.

link-management— (TX Matrix and TX Matrix Plus routers and EX Series switches only) (Optional) Restart the Link Management Protocol (LMP) process, which establishes and maintains LMP control channels.

lldpd-service—(EX Series switches only) (Optional) Restart the Link Layer Discovery Protocol (LLDP) process.

local—(MX Series routers only) (Optional) Restart the software process for the local Virtual Chassis member.

local-policy-decision-function— (Optional) Restart the process for the Local Policy Decision Function, which regulates collection of statistics related to applications and application groups and tracking of information about dynamic subscribers and static interfaces.

mac-validation— (Optional) Restart the Media Access Control (MAC) validation process, which configures MAC address validation for subscriber interfaces created on demux interfaces in dynamic profiles on MX Series routers.

member *member-id*—(MX Series routers only) (Optional) Restart the software process for a specific member of the Virtual Chassis configuration. Replace *member-id* with a value of 0 or 1.

mib-process—(Optional) Restart the Management Information Base (MIB) version II process, which provides the router's MIB II agent.

mobile-ip—(Optional) Restart the Mobile IP process, which configures Junos OS Mobile IP features.

mountd-service—(EX Series switches and MX Series routers only) (Optional) Restart the service for NFS mount requests.

mpls-traceroute—(Optional) Restart the MPLS Periodic Traceroute process.

mspd—(Optional) Restart the Multiservice process.

multicast-snooping—(EX Series switches and MX Series routers only) (Optional) Restart the multicast snooping process, which makes Layer 2 devices, such as VLAN switches, aware of Layer 3 information, such as the media access control (MAC) addresses of members of a multicast group.

named-service—(Optional) Restart the DNS Server process, which is used by a router or a switch to resolve hostnames into addresses.

network-access-service—(J Series routers and QFX Series only) (Optional) Restart the network access process, which provides the router's Challenge Handshake Authentication Protocol (CHAP) authentication service.

nfsd-service—(Optional) Restart the Remote NFS Server process, which provides remote file access for applications that need NFS-based transport.

packet-triggered-subscribers—(Optional) Restart the packet-triggered subscribers and policy control (PTSP) process, which allows the application of policies to dynamic subscribers that are controlled by a subscriber termination device.

peer-selection-service—(Optional) Restart the Peer Selection Service process.

pgcp-service—(Optional) Restart the pgcpd service process running on the Routing Engine. This option does not restart pgcpd processes running on mobile station PICs. To restart pgcpd processes running on mobile station PICs, use the **services pgcp gateway** option.

pgm—(Optional) Restart the process that implements the Pragmatic General Multicast (PGM) protocol for assisting in the reliable delivery of multicast packets.

pic-services-logging—(Optional) Restart the logging process for some PICs. With this process, also known as fsad (the file system access daemon), PICs send special logging information to the Routing Engine for archiving on the hard disk.

pki-service—(Optional) Restart the PKI Service process.

ppp—(Optional) Restart the Point-to-Point Protocol (PPP) process, which is the encapsulation protocol process for transporting IP traffic across point-to-point links.

ppp-service—(Optional) Restart the Universal Edge PPP process, which is the encapsulation protocol process for transporting IP traffic across Universal Edge routers.

pppoe—(Optional) Restart the Point-to-Point Protocol over Ethernet (PPPoE) process, which combines PPP that typically runs over broadband connections with the Ethernet link-layer protocol that allows users to connect to a network of hosts over a bridge or access concentrator.

protected-system-domain-service—(Optional) Restart the Protected System Domain (PSD) process.

redundancy-interface-process—(Optional) Restart the ASP redundancy process.

remote-operations—(Optional) Restart the remote operations process, which provides the ping and traceroute MIBs.

root-system-domain-service—(Optional) Restart the Root System Domain (RSD) service.

routing—(ACX Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the routing protocol process.

routing <logical-system *logical-system-name*>—(Optional) Restart the routing protocol process, which controls the routing protocols that run on the router or switch and maintains the routing tables. Optionally, restart the routing protocol process for the specified logical system only.

sampling—(Optional) Restart the sampling process, which performs packet sampling based on particular input interfaces and various fields in the packet header.

sbcc-configuration-process—(Optional) Restart the session border controller (SBC) process of the border signaling gateway (BSG).

scc—(TX Matrix routers only) (Optional) Restart the software process on the TX Matrix router (or switch-card chassis).

sdk-service—(Optional) Restart the SDK Service process, which runs on the Routing Engine and is responsible for communications between the SDK application and Junos OS. Although the SDK Service process is present on the router, it is turned off by default.

secure-neighbor-discovery—(QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the secure Neighbor Discovery Protocol (NDP) process, which provides support for protecting NDP messages.

sfc number—(TX Matrix Plus routers only) (Optional) Restart the software process on the TX Matrix Plus router (or switch-fabric chassis). Replace **number** with **0**.

service-deployment—(Optional) Restart the service deployment process, which enables Junos OS to work with the Session and Resource Control (SRC) software.

services—(Optional) Restart a service.

services pgcp gateway gateway-name—(Optional) Restart the pgcpd process for a specific border gateway function (BGF) running on an MS-PIC. This option does not restart the pgcpd process running on the Routing Engine. To restart the pgcpd process on the Routing Engine, use the **pgcp-service** option.

sflow-service—(EX Series switches only) (Optional) Restart the flow sampling (sFlow technology) process.

snmp—(Optional) Restart the SNMP process, which enables the monitoring of network devices from a central location and provides the router's or switch's SNMP master agent.

soft—(Optional) Reread and reactivate the configuration without completely restarting the software processes. For example, BGP peers stay up and the routing table stays constant. Omitting this option results in a graceful restart of the software process.

static-subscribers—(Optional) Restart the static subscribers process, which associates subscribers with statically configured interfaces and provides dynamic service activation and activation for these subscribers.

statistics-service—(Optional) Restart the process that manages the Packet Forwarding Engine statistics.

subscriber-management—(Optional) Restart the Subscriber Management process.

subscriber-management-helper—(Optional) Restart the Subscriber Management Helper process.

tunnel-oamd—(Optional) Restart the Tunnel OAM process, which enables the Operations, Administration, and Maintenance of Layer 2 tunneled networks. Layer 2 protocol

tunneling (L2PT) allows service providers to send Layer 2 PDUs across the provider's cloud and deliver them to Juniper Networks EX Series Ethernet Switches that are not part of the local broadcast domain.

usb-control—(J Series routers and MX Series routers only) (Optional) Restart the USB control process.

vrrp—(ACX Series routers, EX Series switches, and MX Series routers only) (Optional) Restart the Virtual Router Redundancy Protocol (VRRP) process, which enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts.

web-management—(J Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the Web management process.

Required Privilege Level

reset

Related Documentation

- Overview of Junos OS CLI Operational Mode Commands

List of Sample Output [restart interfaces on page 147](#)

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

[restart interfaces](#)

```
user@host> restart interfaces
interfaces process terminated
interfaces process restarted
```

show policy

Syntax	show policy <logical-system (all <i>logical-system-name</i>)> < <i>policy-name</i> >
Syntax (EX Series Switches)	show policy < <i>policy-name</i> >
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display information about configured routing policies.
Options	<p>none—List the names of all configured routing policies.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>policy-name</i>—(Optional) Show the contents of the specified policy.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> show policy damping
List of Sample Output	show policy on page 149 show policy policy-name on page 149 show policy (Multicast Scoping) on page 149
Output Fields	Table 6 on page 148 lists the output fields for the show policy command. Output fields are listed in the approximate order in which they appear.

Table 6: show policy Output Fields

Field Name	Field Description
<i>policy-name</i>	Name of the policy listed.
<i>term</i>	Policy term listed.
<i>from</i>	Match condition for the policy.
<i>then</i>	Action for the policy.

Sample Output

show policy

```
user@host> show policy
Configured policies:
__vrf-export-red-internal__
__vrf-import-red-internal__
red-export
all_routes
```

**show policy
policy-name**

```
user@host> show policy test-statics
Policy test-statics:
  from
    3.0.0.0/8  accept
    3.1.0.0/16  accept
  then reject
```

**show policy (Multicast
Scoping)**

```
user@host> show policy test-statics
Policy test-statics:
  from
    multicast-scoping == 8
```

show policy conditions

Syntax	<pre>show policy conditions <condition-name> <detail> <dynamic> <logical-system (all logical-system-name)></pre>
Syntax (EX Series Switches)	<pre>show policy conditions <condition-name> <detail> <dynamic></pre>
Release Information	<p>Command introduced in Junos OS Release 9.0.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Display all the configured conditions as well as the routing tables with which the configuration manager is interacting. If the detail keyword is included, the output also displays dependent routes for each condition.</p>
Options	<p>none—Display all configured conditions and associated routing tables.</p> <p>condition-name—(Optional) Display information about the specified condition only.</p> <p>detail—(Optional) Display the specified level of output.</p> <p>dynamic—(Optional) Display information about the conditions in the dynamic database.</p> <p>logical-system (all logical-system-name)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show policy conditions detail on page 151
Output Fields	<p>Table 7 on page 150 lists the output fields for the show policy conditions command. Output fields are listed in the approximate order in which they appear.</p>

Table 7: show policy conditions Output Fields

Field Name	Field Description	Level of Output
Condition	Name of configured condition.	All levels
event	Condition type. If the if-route-exists option is configured, the event type is: Existence of a route in a specific routing table.	All levels
Dependent routes	List of routes dependent on the condition, along with the latest generation number.	detail
Condition tables	List of routing tables associated with the condition, along with the latest generation number and number of dependencies.	All levels

Table 7: show policy conditions Output Fields (*continued*)

Field Name	Field Description	Level of Output
If-route-exists conditions	List of conditions configured to look for a route in the specified table.	All levels

Sample Output

```
show policy conditions detail user@host> show policy conditions detail
Configured conditions:
Condition cond1, event: Existence of a route in a specific routing table
Dependent routes:
  4.4.4.4/32, generation 3
  6.6.6.6/32, generation 3
  10.10.10.10/32, generation 3

Condition cond2, event: Existence of a route in a specific routing table
Dependent routes:
None

Condition tables:
Table inet.0, generation 4, dependencies 3, If-route-exists conditions: cond1
(static) cond2 (static)
```

show rip general-statistics

Syntax	show rip general-statistics <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches and QFX Series)	show rip general-statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series.
Description	Display brief RIP statistics.
Options	<p>none—Display brief RIP statistics.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> clear rip general-statistics on page 136
List of Sample Output	show rip general-statistics on page 153
Output Fields	<p>Table 8 on page 152 lists the output fields for the show rip general-statistics command. Output fields are listed in the approximate order in which they appear.</p>

Table 8: show rip general-statistics Output Fields

Field Name	Field Description
bad msgs	Number of invalid messages received.
no rcv intf	Number of packets received with no matching interface.
curr memory	Amount of memory currently used by RIP.
max memory	Most memory used by RIP.

Sample Output

```
show rip
general-statistics

user@host> show rip general-statistics
RIPv2 I/O info:
  bad msgs      :      0
  no recv intf  :      0
  curr memory   :      0
  max memory    :      0
```

show rip neighbor

Syntax	show rip neighbor <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> < <i>name</i> >
Syntax (EX Series Switches and QFX Series)	show rip neighbor <instance (all <i>instance-name</i>)> < <i>name</i> >
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series.
Description	Display information about RIP neighbors.
Options	<p>none—Display information about all RIP neighbors for all instances.</p> <p>instance (all <i>instance-name</i>)—(Optional) Display RIP neighbor information for all instances or for only the specified routing instance.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>name</i>—(Optional) Display detailed information about only the specified RIP neighbor.</p>
Required Privilege Level	view
List of Sample Output	show rip neighbor on page 155 show rip neighbor (With Demand Circuits Configured) on page 155
Output Fields	Table 9 on page 154 lists the output fields for the show rip neighbor command. Output fields are listed in the approximate order in which they appear.

Table 9: show rip neighbor Output Fields

Field Name	Field Description
Neighbor	<p>Name of the RIP neighbor.</p> <p>NOTE: Beginning with Junos OS Release 11.1, when you configure demand circuits, the output displays a demand circuit (DC) flag next to neighbor interfaces configured for demand circuits.</p> <p>If you configure demand circuits at the [edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>] hierarchy level, the output shows only the neighboring interface that you specifically configured as a demand circuit. If you configure demand circuits at the [edit protocols rip group <i>group-name</i>] hierarchy level, all of the interfaces in the group are configured as demand circuits. Therefore, the output shows all of the interfaces in that group as demand circuits.</p>

Table 9: show rip neighbor Output Fields (*continued*)

Field Name	Field Description
State	State of the connection: Up or Dn (Down).
Source Address	Address of the port on the local router.
Destination Address	Address of the port on the remote router.
Send Mode	Send options: broadcast , multicast , none , or version 1 .
Receive Mode	Type of packets to accept: both , none , version 1 , or version 2 .
In Met	Metric added to incoming routes when advertising into RIP routes that were learned from other protocols.

Sample Output

show rip neighbor

```

user@host> show rip neighbor
Neighbor      Local State Source Address Destination Address Send Mode Receive Mode In Met
-----
ge-2/3/0.0    Up  192.168.9.105 192.168.9.107 bcast both 1
at-5/1/1.42   Dn  (null)        (null)        mcast v2 only 3
at-5/1/0.42   Dn  (null)        (null)        mcast both 3
at-5/1/0.0    Up  20.0.0.1      224.0.0.9     mcast both 3
so-0/0/0.0    Up  192.168.9.97  224.0.0.9     mcast both 3

```

show rip neighbor (With Demand Circuits Configured)

```

user@host> show rip neighbor
Neighbor      Local State Source Address Destination Address Send Mode Receive Mode In Met
-----
so-0/1/0.0(DC) Up  10.10.10.2    224.0.0.9     mcast both 1
so-0/2/0.0(DC) Up  13.13.13.2    224.0.0.9     mcast both 1

```

show rip statistics

Syntax	show rip statistics <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> <name> <peer (all <i>address</i>)>
Syntax (EX Series Switches and QFX Series)	show rip statistics <instance (all <i>instance-name</i>)> <name>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series.
Description	Display RIP statistics about messages sent and received on an interface, as well as information received from advertisements from other routing devices.
Options	none —Display RIP statistics for all routing instances. instance (all <i>instance-name</i>) —(Optional) Display RIP statistics for all instances or for only the specified routing instance. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system. name —(Optional) Display detailed information about only the specified RIP neighbor. peer (all <i>address</i>) —(Optional) Display RIP statistics for a single peer or all peers.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• clear rip statistics on page 137
List of Sample Output	show rip statistics on page 158
Output Fields	Table 10 on page 157 lists the output fields for the show rip statistics command. Output fields are listed in the approximate order in which they appear.

Table 10: show rip statistics Output Fields

Field Name	Field Description
RIP info	<p>Information about RIP on the specified interface:</p> <ul style="list-style-type: none"> • port—UDP port number used for RIP. • update interval—Interval between routing table updates, in seconds. • holddown—Hold-down interval, in seconds. • timeout—Timeout interval, in seconds. • restart in progress—Graceful restart status. Displayed when RIP is or has been in the process of graceful restart. • restart time—Estimated time for the graceful restart to finish, in seconds. • restart will complete in—Remaining time for the graceful restart to finish, in seconds. • rts learned—Number of routes learned through RIP. • rts held down—Number of routes held down by RIP. • rqsts dropped—Number of received request packets that were dropped. • resps dropped—Number of received response packets that were dropped.
<i>logical-interface</i>	<p>Name of the logical interface and its statistics:</p> <ul style="list-style-type: none"> • routes learned—Number of routes learned on the logical interface. • routes advertised—Number of routes advertised by the logical interface.
Counter	<p>List of counter types:</p> <ul style="list-style-type: none"> • Updates Sent—Number of update messages sent. • Triggered Updates Sent—Number of triggered update messages sent. • Responses Sent—Number of response messages sent. • Bad Messages—Number of invalid messages received. • RIPv1 Updates Received—Number of RIPv1 update messages received. • RIPv1 Bad Route Entries—Number of RIPv1 invalid route entry messages received. • RIPv1 Updates Ignored—Number of RIPv1 update messages ignored. • RIPv2 Updates Received—Number of RIPv2 update messages received. • RIPv2 Bad Route Entries—Number of RIPv2 invalid route entry messages received. • RIPv2 Updates Ignored—Number of RIPv2 update messages ignored. • Authentication Failures—Number of received update messages that failed authentication. • RIP Requests Received—Number of RIP request messages received. • RIP Requests Ignored—Number of RIP request messages ignored.
Total	Total number of packets for the selected counter.
Last 5 min	Number of packets for the selected counter in the most recent 5-minute period.
Last minute	Number of packets for the selected counter in the most recent 1-minute period.

Sample Output

show rip statistics

```
user@host> show rip statistics so-0/0/0.0
RIP info: port 520; update interval: 30s; holddown 180s; timeout 120s
restart in progress: restart time 60s; restart will complete in 55s
    rts learned  rts held down  rqsts dropped  resps dropped
              0              0              0              0
so-0/0/0.0:  0 routes learned; 501 routes advertised
Counter      Total      Last 5 min  Last minute
-----
Updates Sent          0          0          0
Triggered Updates Sent  0          0          0
Responses Sent        0          0          0
Bad Messages          0          0          0
RIPv1 Updates Received  0          0          0
RIPv1 Bad Route Entries  0          0          0
RIPv1 Updates Ignored   0          0          0
RIPv2 Updates Received  0          0          0
RIPv2 Bad Route Entries  0          0          0
RIPv2 Updates Ignored   0          0          0
Authentication Failures  0          0          0
RIP Requests Received   0          0          0
RIP Requests Ignored    0          0          0
```

show route

Syntax	<pre>show route <all> <destination-prefix> <logical-system (all logical-system-name)> <private></pre>
Syntax (EX Series Switches)	<pre>show route <all> <destination-prefix> <private></pre>
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option private introduced in Junos OS Release 9.5.</p> <p>Option private introduced in Junos OS Release 9.5 for EX Series switches.</p>
Description	Display the active entries in the routing tables.
Options	<p>none—Display brief information about all active entries in the routing tables.</p> <p>all—(Optional) Display information about all routing tables, including private, or internal, routing tables.</p> <p>destination-prefix—(Optional) Display active entries for the specified address or range of addresses.</p> <p>logical-system (all logical-system-name)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p>private—(Optional) Display information only about all private, or internal, routing tables.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 13 • Example: Configuring RIPng • Example: Configuring IS-IS • Examples: Configuring Internal BGP Peering • Examples: Configuring External BGP Peering • Examples: Configuring OSPF Routing Policy
List of Sample Output	<p>show route on page 163</p> <p>show route destination-prefix on page 163</p> <p>show route extensive on page 163</p>

Output Fields Table 11 on page 160 describes the output fields for the **show route** command. Output fields are listed in the approximate order in which they appear.

Table 11: show route Output Fields

Field Name	Field Description
<i>routing-table-name</i>	Name of the routing table (for example, inet.0).
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.
<i>number routes</i>	<p>Number of routes in the routing table and total number of routes in the following states:</p> <ul style="list-style-type: none"> • active (routes that are active). • holddown (routes that are in the pending state before being declared inactive). A holddown route was once the active route and is no longer the active route. The route is in the holddown state because a protocol still has interest in the route, meaning that the interest bit is set. A protocol might have its interest bit set on the previously active route because the protocol is still advertising the route. The route will be deleted after all protocols withdraw their advertisement of the route and remove their interest bit. A persistent holddown state often means that the interested protocol is not releasing its interest bit properly. <p>However, if you have configured advertisement of multiple routes (with the add-path or advertise-inactive statement), the holddown bit is most likely set because BGP is advertising the route as an active route. In this case, you can ignore the holddown state because nothing is wrong.</p> <ul style="list-style-type: none"> • hidden (routes that are not used because of a routing policy).
<i>destination-prefix</i>	<p>Route destination (for example:10.0.0.1/24). Sometimes the route information is presented in another format, such as:</p> <ul style="list-style-type: none"> • MPLS-label (for example, 80001). • interface-name (for example, ge-1/0/2). • neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only. For example, 10.1.1.195:NoCtrlWord:1:1:Local/96): <ul style="list-style-type: none"> • neighbor-address—Address of the neighbor. • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord. • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport. • vc-id—Virtual circuit identifier. • source—Source of the advertisement: Local or Remote.
[<i>protocol, preference</i>]	<p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • -—A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route. <p>In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value and a lower Preference2 value.</p>

Table 11: show route Output Fields (*continued*)

Field Name	Field Description
<i>weeks:days</i> <i>hours:minutes:seconds</i>	How long the route been known (for example, 2w4d 13:11:14 , or 2 weeks, 4 days, 13 hours, 11 minutes, and 14 seconds).
metric	Cost value of the indicated route. For routes within an AS, the cost is determined by the IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.
localpref	Local preference value included in the route.
from	Interface from which the route was received.
AS path	<p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device, or if AS path prepending is configured. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p>
validation-state	<p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> • Invalid—Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • Unknown—Indicates that the prefix is not among the prefixes or prefix ranges in the database. • Valid—Indicates that the prefix and autonomous system pair are found in the database.
to	<p>Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.</p> <p>If the destination is Discard, traffic is dropped.</p>

Table 11: show route Output Fields (*continued*)

Field Name	Field Description
via	<p>Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word Selected. This field can also contain the following information:</p> <ul style="list-style-type: none">• Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible.• Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.• lsp-path-name—Name of the LSP used to reach the next hop.• label-action—MPLS label and operation occurring at the next hop. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label).

Sample Output

show route

```
user@host> show route
inet.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:65500:1:10.0.0.20/240
    *[MVPN/70] 19:53:41, metric2 1
    Indirect
1:65500:1:10.0.0.40/240
    *[BGP/170] 19:53:29, localpref 100, from 10.0.0.30
    AS path: I
    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
    [BGP/170] 19:53:26, localpref 100, from 10.0.0.33
    AS path: I
    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
1:65500:1:10.0.0.60/240
    *[BGP/170] 19:53:29, localpref 100, from 10.0.0.30
    AS path: I
    > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF
    [BGP/170] 19:53:25, localpref 100, from 10.0.0.33
    AS path: I
    > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF
```

show route destination-prefix

```
user@host> show route 172.16.0.0/12

inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.0/12    *[Static/5] 2w4d 12:54:27
    > to 192.168.167.254 via fxp0.0
```

show route extensive

```
user@host> show route extensive
v1.mvpn.0: 5 destinations, 8 routes (5 active, 1 holddown, 0 hidden)
1:65500:1:10.0.0.40/240 (1 entry, 1 announced)
    *BGP Preference: 170/-101
    PMSI: Flags 0x0: Label[0:0:0]: PIM-SM: Sender 10.0.0.40 Group 225.1.1.1

    Next hop type: Indirect
    Address: 0x92455b8
    Next-hop reference count: 2
    Source: 10.0.0.30
    Protocol next hop: 10.0.0.40
    Indirect next hop: 2 no-forward
    State: <Active Int Ext>
        Local AS: 65500 Peer AS: 65500
    Age: 3 Metric2: 1
    Task: BGP_65500.10.0.0.30+179
    Announcement bits (2): 0-PIM.v1 1-mvpn global task
    AS path: I (Originator) Cluster list: 10.0.0.30
    AS path: Originator ID: 10.0.0.40
    Communities: target:65520:100
    Import Accepted
    Localpref: 100
    Router ID: 10.0.0.30
    Primary Routing Table bgp.mvpn.0
    Indirect next hops: 1
        Protocol next hop: 10.0.0.40 Metric: 1
```

```
Indirect next hop: 2 no-forward
Indirect path forwarding next hops: 1
  Next hop type: Router
  Next hop: 10.0.24.4 via lt-0/3/0.24 weight 0x1
10.0.0.40/32 Originating RIB: inet.3
  Metric: 1                      Node path count: 1
  Forwarding nexthops: 1
    Nexthop: 10.0.24.4 via lt-0/3/0.24
```

show route active-path

Syntax	show route active-path <brief detail extensive terse> <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches)	show route active-path <brief detail extensive terse>
Release Information	Command introduced in Junos OS Release 8.0. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display all active routes for destinations. An active route is a route that is selected as the best path. Inactive routes are not displayed.
Options	<p>none—Display all active routes.</p> <p>brief detail extensive terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route active-path on page 166 show route active-path brief on page 166 show route active-path detail on page 166 show route active-path extensive on page 167 show route active-path terse on page 169
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route active-path user@host> **show route active-path**

```
inet.0: 7 destinations, 7 routes (6 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.70.19/32    *[Direct/0] 21:33:52
                  > via lo0.0
10.255.71.50/32    *[IS-IS/15] 00:18:13, metric 10
                  > to 100.1.2.1 via so-2/1/3.0
100.1.2.0/24       *[Direct/0] 00:18:36
                  > via so-2/1/3.0
100.1.2.2/32       *[Local/0] 00:18:41
                  Local via so-2/1/3.0
192.168.64.0/21    *[Direct/0] 21:33:52
                  > via fxp0.0
192.168.70.19/32   *[Local/0] 21:33:52
                  Local via fxp0.0
```

show route active-path brief The output for the **show route active-path brief** command is identical to that for the **show route active-path** command. For sample output, see [show route active-path on page 166](#).

show route active-path detail user@host> **show route active-path detail**

```
inet.0: 7 destinations, 7 routes (6 active, 0 holddown, 1 hidden)

10.255.70.19/32 (1 entry, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
    Next hop: via lo0.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 21:37:10
    Task: IF
    Announcement bits (3): 2-IS-IS 5-Resolve tree 2 6-Resolve tree 3
    AS path: I

10.255.71.50/32 (1 entry, 1 announced)
  *IS-IS Preference: 15
    Level: 1
    Next hop type: Router, Next hop index: 397
    Next-hop reference count: 4
    Next hop: 100.1.2.1 via so-2/1/3.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 21:31 Metric: 10
    Task: IS-IS
    Announcement bits (4): 0-KRT 2-IS-IS 5-Resolve tree 2 6-Resolve
tree 3
    AS path: I

100.1.2.0/24 (1 entry, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
```

```

Next hop: via so-2/1/3.0, selected
State: <Active Int>
Local AS: 200
Age: 21:54
Task: IF
Announcement bits (3): 2-IS-IS 5-Resolve tree 2 6-Resolve tree 3

AS path: I

100.1.2.2/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 11
    Interface: so-2/1/3.0
    State: <Active NoReadvrt Int>
    Local AS: 200
    Age: 21:59
    Task: IF
    Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
    AS path: I

192.168.64.0/21 (1 entry, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
    Next hop: via fxp0.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 21:37:10
    Task: IF
    Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
    AS path: I

192.168.70.19/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 11
    Interface: fxp0.0
    State: <Active NoReadvrt Int>
    Local AS: 200
    Age: 21:37:10
    Task: IF
    Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
    AS path: I

```

show route active-path extensive

```

user@host> show route active-path extensive

inet.0: 7 destinations, 7 routes (6 active, 0 holddown, 1 hidden)
10.255.70.19/32 (1 entry, 1 announced)
TSI:
IS-IS level 1, LSP fragment 0
IS-IS level 2, LSP fragment 0
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
    Next hop: via lo0.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 21:39:47
    Task: IF
    Announcement bits (3): 2-IS-IS 5-Resolve tree 2 6-Resolve tree 3

```

```

AS path: I

10.255.71.50/32 (1 entry, 1 announced)
TSI:
KRT in-kerne1 10.255.71.50/32 -> {100.1.2.1}
IS-IS level 2, LSP fragment 0
  *IS-IS Preference: 15
    Level: 1
    Next hop type: Router, Next hop index: 397
    Next-hop reference count: 4
    Next hop: 100.1.2.1 via so-2/1/3.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 24:08 Metric: 10
    Task: IS-IS
    Announcement bits (4): 0-KRT 2-IS-IS 5-Resolve tree 2 6-Resolve
tree 3
AS path: I

100.1.2.0/24 (1 entry, 1 announced)
TSI:
IS-IS level 1, LSP fragment 0
IS-IS level 2, LSP fragment 0
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
    Next hop: via so-2/1/3.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 24:31
    Task: IF
    Announcement bits (3): 2-IS-IS 5-Resolve tree 2 6-Resolve tree 3

AS path: I

100.1.2.2/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 11
    Interface: so-2/1/3.0
    State: <Active NoReadvrt Int>
    Local AS: 200
    Age: 24:36
    Task: IF
    Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
AS path: I

192.168.64.0/21 (1 entry, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 3
    Next hop: via fxp0.0, selected
    State: <Active Int>
    Local AS: 200
    Age: 21:39:47
    Task: IF
    Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
AS path: I

192.168.70.19/32 (1 entry, 1 announced)

```



```

*Local Preference: 0
  Next hop type: Local
  Next-hop reference count: 11
  Interface: fxp0.0
  State: <Active NoReadvrt Int>
  Local AS: 200
  Age: 21:39:47
  Task: IF
  Announcement bits (2): 5-Resolve tree 2 6-Resolve tree 3
  AS path: I

```

show route active-path terse user@host> **show route active-path terse**

```

inet.0: 7 destinations, 7 routes (6 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

```

A	Destination	P	Prf	Metric 1	Metric 2	Next hop	AS path
*	10.255.70.19/32	D	0			>lo0.0	
*	10.255.71.50/32	I	15	10		>100.1.2.1	
*	100.1.2.0/24	D	0			>so-2/1/3.0	
*	100.1.2.2/32	L	0			Local	
*	192.168.64.0/21	D	0			>fxp0.0	
*	192.168.70.19/32	L	0			Local	

show route advertising-protocol

Syntax	<code>show route advertising-protocol <i>protocol</i> <i>neighbor-address</i></code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Release Information	Command introduced before Junos OS Release 7.4.
Description	Display the routing information as it has been prepared for advertisement to a particular neighbor of a particular dynamic routing protocol.
Options	<p>brief detail extensive terse—(Optional) Display the specified level of output.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>neighbor-address</i>—Address of the neighboring router to which the route entry is being transmitted.</p> <p><i>protocol</i>—Protocol transmitting the route:</p> <ul style="list-style-type: none"> • bgp—Border Gateway Protocol • dvmrp—Distance Vector Multicast Routing Protocol • msdp—Multicast Source Discovery Protocol • pim—Protocol Independent Multicast • rip—Routing Information Protocol • ripng—Routing Information Protocol next generation
Additional Information	Routes displayed are routes that the routing table has exported into the routing protocol and that have been filtered by the associated protocol's export routing policy statements.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the MED Attribute Directly
List of Sample Output	show route advertising-protocol bgp (Layer 3 VPN) on page 173 show route advertising-protocol bgp detail on page 173 show route advertising-protocol bgp detail (Layer 2 VPN) on page 173 show route advertising-protocol bgp detail (Layer 3 VPN) on page 173 show route advertising-protocol bgp extensive all (Next Hop Self with RIB-out IP Address) on page 174
Output Fields	Table 12 on page 171 lists the output fields for the show route advertising-protocol command. Output fields are listed in the approximate order in which they appear.

Table 12: show route advertising-protocol Output Fields

Field Name	Field Description	Level of Output
<i>routing-table-name</i>	Name of the routing table—for example, inet.0.	All levels
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.	All levels
<i>number routes</i>	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> • active (routes that are active) • holddown (routes that are in the pending state before being declared inactive) • hidden (routes that are not used because of a routing policy) 	All levels
Prefix	Destination prefix.	brief none
<i>destination-prefix (entry, announced)</i>	Destination prefix. The entry value is the number of routes for this destination, and the announced value is the number of routes being announced for this destination.	detail extensive
BGP group and type	BGP group name and type (Internal or External).	detail extensive
Route Distinguisher	Unique 64-bit prefix augmenting each IP subnet.	detail extensive
Advertised Label	Incoming label advertised by the LDP. When an IP packet enters a label-switched path (LSP), the ingress router examines the packet and assigns it a label based on its destination, placing the label in the packet's header. The label transforms the packet from one that is forwarded based on its IP routing information to one that is forwarded based on information associated with the label.	detail extensive
Label-Base, range	First label in a block of labels and label block size. A remote PE router uses this first label when sending traffic toward the advertising PE router.	detail extensive
VPN Label	Virtual private network (VPN) label. Packets are sent between CE and PE routers by advertising VPN labels. VPN labels transit over either an RSVP or an LDP LSP tunnel.	detail extensive
Nexthop	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route. If the next-hop advertisement to the peer is Self , and the RIB-out next hop is a specific IP address, the RIB-out IP address is included in the extensive output. See show route advertising-protocol bgp extensive all (Next Hop Self with RIB-out IP Address) on page 174.	All levels
MED	Multiple exit discriminator value included in the route.	brief
Lclpref or Localpref	Local preference value included in the route.	All levels

Table 12: show route advertising-protocol Output Fields (*continued*)

Field Name	Field Description	Level of Output
AS path	<p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the local AS number associated with the AS path if configured on the router, or if AS path prepending is configured. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p>	All levels
Communities	Community path attribute for the route. See the output field table for the show route detail command for all possible values for this field.	detail extensive
AIGP	Accumulated interior gateway protocol (AIGP) BGP attribute.	detail extensive
Attrset AS	Number, local preference, and path of the autonomous system (AS) that originated the route. These values are stored in the Attrset attribute at the originating router.	detail extensive
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).	detail extensive
control flags	Control flags: none or Site Down .	detail extensive
mtu	Maximum transmission unit (MTU) of the Layer 2 circuit.	detail extensive

Sample Output

**show route
advertising-protocol
bgp (Layer 3 VPN)**

```
user@host> show route advertising-protocol bgp 10.255.14.171
VPN-A.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED    Lclpref AS path
10.255.14.172/32 Self              1      100 I
VPN-B.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED    Lclpref AS path
10.255.14.181/32 Self              2      100 I
```

**show route
advertising-protocol
bgp detail**

```
user@host> show route advertising-protocol bgp 111.222.1.3 detail
bgp20.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
111.222.1.11/32 (1 entry, 1 announced)
  BGP group pe-pe type Internal
    Route Distinguisher: 111.255.14.11:69
    Advertised Label: 100000
    next hop: Self
    Localpref: 100
    AS path: 2 I
    Communities: target:69:20
    AIGP 210
111.8.0.0/16 (1 entry, 1 announced)
  BGP group pe-pe type Internal
    Route Distinguisher: 111.255.14.11:69
    Advertised Label: 100000
    Next hop: Self
    Localpref: 100
    AS path: 2 I
    Communities: target:69:20
    AIGP 210
```

**show route
advertising-protocol
bgp detail (Layer 2
VPN)**

```
user@host> show route advertising-protocol bgp 192.168.24.1 detail
vpn-a.12vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
192.168.16.1:1:1:1/96 (1 entry, 1 announced)
  BGP group int type Internal
    Route Distinguisher: 192.168.16.1:1
    Label-base : 32768, range : 3
    Nexthop: Self
    Localpref: 100
    AS path: I
    Communities: target:65412:100
    AIGP 210
    Layer2-info: encaps:VLAN, control flags:, mtu:
```

**show route
advertising-protocol**

```
user@host> show route advertising-protocol bgp 10.255.14.176 detail
vpna.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
* 10.49.0.0/30 (1 entry, 1 announced)
```

**bgp detail (Layer 3
VPN)**

```
BGP group ibgp type Internal
Route Distinguisher: 10.255.14.174:2
VPN Label: 101264
Nexthop: Self
Localpref: 100
AS path: I
Communities: target:200:100
AIGP 210
AttrSet AS: 100
    Localpref: 100
    AS path: I
...
```

**show route
advertising-protocol
bgp extensive all (Next
Hop Self with RIB-out
IP Address)**

```
user@host> show route advertising-protocol bgp 200.0.0.2 170.0.1.0/24 extensive all
inet.0: 13 destinations, 19 routes (13 active, 0 holddown, 6 hidden)
  170.0.1.0/24 (2 entries, 1 announced)
    BGP group eBGP-INTEROP type External
      Nexthop: Self (rib-out 10.100.3.2)
      AS path: [4713] 200 I
...
```

show route all

Syntax	show route all <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches)	show route all
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display information about all routes in all routing tables, including private, or internal, tables.
Options	none —Display information about all routes in all routing tables, including private, or internal, tables. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	show route all on page 176
Output Fields	In Junos OS Release 9.5 and later, only the output fields for the show route all command display all routing tables, including private, or hidden, routing tables. The output field table of the show route command does not display entries for private, or hidden, routing tables in Junos OS Release 9.5 and later.

Sample Output

show route all

The following example displays a snippet of output from the **show route** command and then displays the same snippet of output from the **show route all** command:

```
user@host> show route
mpls.0: 7 destinations, 7 routes (5 active, 0 holddown, 2 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
0          *[MPLS/0] 2d 02:24:39, metric 1
            Receive
1          *[MPLS/0] 2d 02:24:39, metric 1
            Receive
2          *[MPLS/0] 2d 02:24:39, metric 1
            Receive
800017     *[VPLS/7] 1d 14:00:16
            > via vt-3/2/0.32769, Pop
800018     *[VPLS/7] 1d 14:00:26
            > via vt-3/2/0.32772, Pop

user@host> show route all
mpls.0: 7 destinations, 7 routes (5 active, 0 holddown, 2 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
0          *[MPLS/0] 2d 02:19:12, metric 1
            Receive
1          *[MPLS/0] 2d 02:19:12, metric 1
            Receive
2          *[MPLS/0] 2d 02:19:12, metric 1
            Receive
800017     *[VPLS/7] 1d 13:54:49
            > via vt-3/2/0.32769, Pop
800018     *[VPLS/7] 1d 13:54:59
            > via vt-3/2/0.32772, Pop
vt-3/2/0.32769 [VPLS/7] 1d 13:54:49
                Unusable
vt-3/2/0.32772 [VPLS/7] 1d 13:54:59
                Unusable
```


show route best

Syntax	<code>show route best <i>destination-prefix</i></code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Syntax (EX Series Switches)	<code>show route best <i>destination-prefix</i></code> <code><brief detail extensive terse></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display the route in the routing table that is the best route to the specified address or range of addresses. The best route is the longest matching route.
Options	brief detail extensive terse —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief . <i>destination-prefix</i> —Address or range of addresses. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	show route best on page 178 show route best detail on page 178 show route best extensive on page 179 show route best terse on page 179
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route best

```
user@host> show route best 10.255.70.103
inet.0: 24 destinations, 25 routes (23 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
10.255.70.103/32    *[OSPF/10] 1d 13:19:20, metric 2
                  > to 10.31.1.6 via ge-3/1/0.0
                  via so-0/3/0.0

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
10.255.70.103/32    *[RSVP/7] 1d 13:20:13, metric 2
                  > via so-0/3/0.0, label-switched-path green-r1-r3

private1__inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.0.0.0/8          *[Direct/0] 2d 01:43:34
                  > via fxp2.0
                  [Direct/0] 2d 01:43:34
                  > via fxp1.0
```

show route best detail

```
user@host> show route best 10.255.70.103 detail
inet.0: 24 destinations, 25 routes (23 active, 0 holddown, 1 hidden)
Restart Complete
10.255.70.103/32 (1 entry, 1 announced)
    *OSPF    Preference: 10
             Next-hop reference count: 9
             Next hop: 10.31.1.6 via ge-3/1/0.0, selected
             Next hop: via so-0/3/0.0
             State: <Active Int>
             Local AS:    69
             Age: 1d 13:20:06      Metric: 2
             Area: 0.0.0.0
             Task: OSPF
             Announcement bits (2): 0-KRT 3-Resolve tree 2
             AS path: I

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
10.255.70.103/32 (1 entry, 1 announced)
    State: <FlashAll>
    *RSVP    Preference: 7
             Next-hop reference count: 5
             Next hop: via so-0/3/0.0 weight 0x1, selected
             Label-switched-path green-r1-r3
             Label operation: Push 100016
             State: <Active Int>
             Local AS:    69
             Age: 1d 13:20:59      Metric: 2
             Task: RSVP
             Announcement bits (1): 1-Resolve tree 2
             AS path: I

private1__inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
10.0.0.0/8 (2 entries, 0 announced)
    *Direct Preference: 0
             Next hop type: Interface
```

```

Next-hop reference count: 1
Next hop: via fxp2.0, selected
State: <Active Int>
Age: 2d 1:44:20
Task: IF
AS path: I
Direct Preference: 0
Next hop type: Interface
Next-hop reference count: 1
Next hop: via fxp1.0, selected
State: <NotBest Int>
Inactive reason: No difference
Age: 2d 1:44:20
Task: IF
AS path: I

```

show route best extensive

The output for the **show route best extensive** command is identical to that for the **show route best detail** command. For sample output, see [show route best detail on page 178](#).

show route best terse

```

user@host> show route best 10.255.70.103 terse
inet.0: 24 destinations, 25 routes (23 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1   Metric 2   Next hop      AS path
* 10.255.70.103/32  0  10           2           >10.31.1.6
                                     so-0/3/0.0

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1   Metric 2   Next hop      AS path
* 10.255.70.103/32  R   7           2           >so-0/3/0.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1   Metric 2   Next hop      AS path
* 10.0.0.0/8        D   0           0           >fxp2.0
                    D   0           0           >fxp1.0

```

show route brief

Syntax	show route brief <destination-prefix> <logical-system (all logical-system-name)>
Syntax (EX Series Switches)	show route brief <destination-prefix>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display brief information about the active entries in the routing tables.
Options	none —Display all active entries in the routing table. destination-prefix —(Optional) Display active entries for the specified address or range of addresses. logical-system (all logical-system-name) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	show route brief on page 181
Output Fields	For information about output fields, see the Output Field table of the show route command.

Sample Output

show route brief

```

user@host> show route brief
inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 1w5d 20:30:29
                   Discard
10.255.245.51/32   *[Direct/0] 2w4d 13:11:14
                   > via lo0.0
172.16.0.0/12      *[Static/5] 2w4d 13:11:14
                   > to 192.168.167.254 via fxp0.0
192.168.0.0/18     *[Static/5] 1w5d 20:30:29
                   > to 192.168.167.254 via fxp0.0
192.168.40.0/22    *[Static/5] 2w4d 13:11:14
                   > to 192.168.167.254 via fxp0.0
192.168.64.0/18    *[Static/5] 2w4d 13:11:14
                   > to 192.168.167.254 via fxp0.0
192.168.164.0/22   *[Direct/0] 2w4d 13:11:14
                   > via fxp0.0
192.168.164.51/32  *[Local/0] 2w4d 13:11:14
                   Local via fxp0.0
207.17.136.192/32 *[Static/5] 2w4d 13:11:14
                   > to 192.168.167.254 via fxp0.0
green.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100.101.0.0/16     *[Direct/0] 1w5d 20:30:28
                   > via fe-0/0/3.0
100.101.2.3/32     *[Local/0] 1w5d 20:30:28
                   Local via fe-0/0/3.0
224.0.0.5/32       *[OSPF/10] 1w5d 20:30:29, metric 1
                   MultiRecv

```

show route detail

Syntax	show route detail <destination-prefix> <logical-system (all logical-system-name)>
Syntax (EX Series Switches)	show route detail <destination-prefix>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display detailed information about the active entries in the routing tables.
Options	<p>none—Display all active entries in the routing table on all systems.</p> <p>destination-prefix—(Optional) Display active entries for the specified address or range of addresses.</p> <p>logical-system (all logical-system-name)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route detail on page 191 show route detail (with BGP Multipath) on page 196
Output Fields	<p>Table 13 on page 182 describes the output fields for the show route detail command. Output fields are listed in the approximate order in which they appear.</p>

Table 13: show route detail Output Fields

Field Name	Field Description
<i>routing-table-name</i>	Name of the routing table (for example, inet.0).
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.
<i>number routes</i>	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> active (routes that are active) holddown (routes that are in the pending state before being declared inactive) hidden (routes that are not used because of a routing policy)

Table 13: show route detail Output Fields (*continued*)

Field Name	Field Description
<i>route-destination</i> (entry, announced)	<p>Route destination (for example:10.0.0.1/24). The entry value is the number of routes for this destination, and the announced value is the number of routes being announced for this destination. Sometimes the route destination is presented in another format, such as:</p> <ul style="list-style-type: none"> • MPLS-label (for example, 80001). • interface-name (for example, ge-1/0/2). • neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96). <ul style="list-style-type: none"> • neighbor-address—Address of the neighbor. • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord. • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport. • vc-id—Virtual circuit identifier. • source—Source of the advertisement: Local or Remote.
label stacking	<p>(Next-to-the-last-hop routing device for MPLS only) Depth of the MPLS label stack, where the label-popping operation is needed to remove one or more labels from the top of the stack. A pair of routes is displayed, because the pop operation is performed only when the stack depth is two or more labels.</p> <ul style="list-style-type: none"> • S=0 route indicates that a packet with an incoming label stack depth of 2 or more exits this routing device with one fewer label (the label-popping operation is performed). • If there is no S= information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed).
[<i>protocol, preference</i>]	<p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • -—A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route. <p>In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value and a lower Preference2 value.</p>
Level	<p>(IS-IS only). In IS-IS, a single AS can be divided into smaller groups called areas. Routing between areas is organized hierarchically, allowing a domain to be administratively divided into smaller areas. This organization is accomplished by configuring Level 1 and Level 2 intermediate systems. Level 1 systems route within an area. When the destination is outside an area, they route toward a Level 2 system. Level 2 intermediate systems route between areas and toward other ASs.</p>
Route Distinguisher	IP subnet augmented with a 64-bit prefix.
Next-hop type	Type of next hop. For a description of possible values for this field, see Table 14 on page 186 .

Table 13: show route detail Output Fields (*continued*)

Field Name	Field Description
Next-hop reference count	Number of references made to the next hop.
Flood nexthop branches exceed maximum message	Indicates that the number of flood next-hop branches exceeded the system limit of 32 branches, and only a subset of the flood next-hop branches were installed in the kernel.
Source	IP address of the route source.
Next hop	Network layer address of the directly reachable neighboring system.
via	<p>Interface used to reach the next hop. If there is more than one interface available to the next hop, the name of the interface that is actually used is followed by the word Selected. This field can also contain the following information:</p> <ul style="list-style-type: none"> • Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.
Label-switched-path lsp-path-name	Name of the LSP used to reach the next hop.
Label operation	MPLS label and operation occurring at this routing device. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label).
Interface	(Local only) Local interface name.
Protocol next hop	Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop.
Indirect next hop	Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops.
State	State of the route (a route can be in more than one state). See Table 15 on page 188 .
Local AS	AS number of the local routing device.
Age	How long the route has been known.
AIGP	Accumulated interior gateway protocol (AIGP) BGP attribute.
Metricn	Cost value of the indicated route. For routes within an AS, the cost is determined by IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.

Table 13: show route detail Output Fields (*continued*)

Field Name	Field Description
MED-plus-IGP	Metric value for BGP path selection to which the IGP cost to the next-hop destination has been added.
TTL-Action	<p>For MPLS LSPs, state of the TTL propagation attribute. Can be enabled or disabled for all RSVP-signaled and LDP-signaled LSPs or for specific VRF routing instances.</p> <p>For sample output, see show route table.</p>
Task	Name of the protocol that has added the route.
Announcement bits	List of protocols that announce this route. n-Resolve inet indicates that the route is used for route resolution for next hops found in the routing table. n is an index used by Juniper Networks customer support only.
AS path	<p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • Recorded—The AS path is recorded by the sample process (sampled). • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the number that precedes the AS path. This number represents the number of ASs present in the AS path, when calculated as defined in RFC 4271. This value is used in the AS-path merge process, as defined in RFC 4893. • []—If more than one AS number is configured on the routing device, or if AS path prepending is configured, brackets enclose the local AS number associated with the AS path. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p>
VC Label	MPLS label assigned to the Layer 2 circuit virtual connection.
MTU	Maximum transmission unit (MTU) of the Layer 2 circuit.
VLAN ID	VLAN identifier of the Layer 2 circuit.
Prefixes bound to route	Forwarding equivalent class (FEC) bound to this route. Applicable only to routes installed by LDP.
Communities	Community path attribute for the route. See Table 16 on page 190 for all possible values for this field.
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).

Table 13: show route detail Output Fields (*continued*)

Field Name	Field Description
control flags	Control flags: none or Site Down .
mtu	Maximum transmission unit (MTU) information.
Label-Base, range	First label in a block of labels and label block size. A remote PE routing device uses this first label when sending traffic toward the advertising PE routing device.
status vector	Layer 2 VPN and VPLS network layer reachability information (NLRI).
Accepted Multipath	Current active path when BGP multipath is configured.
Accepted MultipathContrib	Path currently contributing to BGP multipath.
Localpref	Local preference value included in the route.
Router ID	BGP router ID as advertised by the neighbor in the open message.
Primary Routing Table	In a routing table group, the name of the primary routing table in which the route resides.
Secondary Tables	In a routing table group, the name of one or more secondary tables in which the route resides.

[Table 14 on page 186](#) describes all possible values for the **Next-hop Types** output field.

Table 14: Next-hop Types Output Field Values

Next-Hop Type	Description
Broadcast (bcast)	Broadcast next hop.
Deny	Deny next hop.
Discard	Discard next hop.
Flood	Flood next hop. Consists of components called branches, up to a maximum of 32 branches. Each flood next-hop branch sends a copy of the traffic to the forwarding interface. Used by point-to-multipoint RSVP, point-to-multipoint LDP, point-to-multipoint CCC, and multicast.
Hold	Next hop is waiting to be resolved into a unicast or multicast type.
Indexed (idxd)	Indexed next hop.

Table 14: Next-hop Types Output Field Values (*continued*)

Next-Hop Type	Description
Indirect (indr)	Used with applications that have a protocol next hop address that is remote. You are likely to see this next-hop type for internal BGP (IBGP) routes when the BGP next hop is a BGP neighbor that is not directly connected.
Interface	Used for a network address assigned to an interface. Unlike the router next hop, the interface next hop does not reference any specific node on the network.
Local (locl)	Local address on an interface. This next-hop type causes packets with this destination address to be received locally.
Multicast (mcst)	Wire multicast next hop (limited to the LAN).
Multicast discard (mdsc)	Multicast discard.
Multicast group (mgrp)	Multicast group member.
Receive (recv)	Receive.
Reject (rjct)	Discard. An ICMP unreachable message was sent.
Resolve (rslv)	Resolving next hop.
Routed multicast (mcrt)	Regular multicast next hop.
Router	<p>A specific node or set of nodes to which the routing device forwards packets that match the route prefix.</p> <p>To qualify as next-hop type router, the route must meet the following criteria:</p> <ul style="list-style-type: none"> • Must not be a direct or local subnet for the routing device. • Must have a next hop that is directly connected to the routing device.
Table	Routing table next hop.
Unicast (ucst)	Unicast.
Unilist (ulst)	List of unicast next hops. A packet sent to this next hop goes to any next hop in the list.

Table 15 on page 188 describes all possible values for the **State** output field. A route can be in more than one state (for example, **<Active NoReadvrt Int Ext>**).

Table 15: State Output Field Values

Value	Description
Accounting	Route needs accounting.
Active	Route is active.
Always Compare MED	Path with a lower multiple exit discriminator (MED) is available.
AS path	Shorter AS path is available.
Cisco Non-deterministic MED selection	Cisco nondeterministic MED is enabled, and a path with a lower MED is available.
Clone	Route is a clone.
Cluster list length	Length of cluster list sent by the route reflector.
Delete	Route has been deleted.
Ex	Exterior route.
Ext	BGP route received from an external BGP neighbor.
FlashAll	Forces all protocols to be notified of a change to any route, active or inactive, for a prefix. When not set, protocols are informed of a prefix only when the active route changes.
Hidden	Route not used because of routing policy.
IfCheck	Route needs forwarding RPF check.
IGP metric	Path through next hop with lower IGP metric is available.
Inactive reason	Flags for this route, which was not selected as best for a particular destination.
Initial	Route being added.
Int	Interior route.
Int Ext	BGP route received from an internal BGP peer or a BGP confederation peer.
Interior > Exterior > Exterior via Interior	Direct, static, IGP, or EBGp path is available.
Local Preference	Path with a higher local preference value is available.
Martian	Route is a martian (ignored because it is obviously invalid).

Table 15: State Output Field Values (*continued*)

Value	Description
MartianOK	Route exempt from martian filtering.
Next hop address	Path with lower metric next hop is available.
No difference	Path from neighbor with lower IP address is available.
NoReadvrt	Route not to be advertised.
NotBest	Route not chosen because it does not have the lowest MED.
Not Best in its group	Incoming BGP AS is not the best of a group (only one AS can be the best).
NotInstall	Route not to be installed in the forwarding table.
Number of gateways	Path with a greater number of next hops is available.
Origin	Path with a lower origin code is available.
Pending	Route pending because of a hold-down configured on another route.
Release	Route scheduled for release.
RIB preference	Route from a higher-numbered routing table is available.
Route Distinguisher	64-bit prefix added to IP subnets to make them unique.
Route Metric or MED comparison	Route with a lower metric or MED is available.
Route Preference	Route with lower preference value is available
Router ID	Path through a neighbor with lower ID is available.
Secondary	Route not a primary route.
Unusable path	Path is not usable because of one of the following conditions: <ul style="list-style-type: none"> • The route is damped. • The route is rejected by an import policy. • The route is unresolved.
Update source	Last tiebreaker is the lowest IP address value.

Table 16 on page 190 describes the possible values for the **Communities** output field.

Table 16: Communities Output Field Values

Value	Description
<i>area-number</i>	4 bytes, encoding a 32-bit area number. For AS-external routes, the value is 0 . A nonzero value identifies the route as internal to the OSPF domain, and as within the identified area. Area numbers are relative to a particular OSPF domain.
<i>bandwidth: local AS number:link-bandwidth-number</i>	Link-bandwidth community value used for unequal-cost load balancing. When BGP has several candidate paths available for multipath purposes, it does not perform unequal-cost load balancing according to the link-bandwidth community unless all candidate paths have this attribute.
<i>domain-id</i>	Unique configurable number that identifies the OSPF domain.
<i>domain-id-vendor</i>	Unique configurable number that further identifies the OSPF domain.
<i>link-bandwidth-number</i>	Link-bandwidth number: from 0 through 4,294,967,295 (bytes per second).
<i>local AS number</i>	Local AS number: from 1 through 65,535 .
<i>options</i>	1 byte. Currently this is only used if the route type is 5 or 7 . Setting the least significant bit in the field indicates that the route carries a type 2 metric.
<i>origin</i>	(Used with VPNs) Identifies where the route came from.
<i>ospf-route-type</i>	1 byte, encoded as 1 or 2 for intra-area routes (depending on whether the route came from a type 1 or a type 2 LSA); 3 for summary routes; 5 for external routes (area number must be 0); 7 for NSSA routes; or 129 for sham link endpoint addresses.
<i>route-type-vendor</i>	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute 0x8000 . The format is <i>area-number:ospf-route-type:options</i> .
<i>rte-type</i>	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute 0x0306 . The format is <i>area-number:ospf-route-type:options</i> .
<i>target</i>	Defines which VPN the route participates in; target has the format <i>32-bit IP address:16-bit number</i> . For example, 10.19.0.0:100.
<i>unknown IANA</i>	Incoming IANA codes with a value between 0x1 and 0x7fff . This code of the BGP extended community attribute is accepted, but it is not recognized.
<i>unknown OSPF vendor community</i>	Incoming IANA codes with a value above 0x8000 . This code of the BGP extended community attribute is accepted, but it is not recognized.

Sample Output

show route detail

user@host> show route detail

```
inet.0: 22 destinations, 23 routes (21 active, 0 holddown, 1 hidden)
10.10.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 29
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 1:31:43
    Task: RT
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

10.31.1.0/30 (2 entries, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 2
    Next hop: via so-0/3/0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:30:17
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I
  OSPF Preference: 10
    Next-hop reference count: 1
    Next hop: via so-0/3/0.0, selected
    State: <Int>
    Inactive reason: Route Preference
    Local AS: 69
    Age: 1:30:17 Metric: 1
    Area: 0.0.0.0
    Task: OSPF
    AS path: I

10.31.1.1/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 7
    Interface: so-0/3/0.0
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:30:20
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I

...

10.31.2.0/30 (1 entry, 1 announced)
  *OSPF Preference: 10
    Next-hop reference count: 9
    Next hop: via so-0/3/0.0
    Next hop: 10.31.1.6 via ge-3/1/0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:29:56 Metric: 2
```

```

Area: 0.0.0.0
Task: OSPF
Announcement bits (2): 0-KRT 3-Resolve tree 2
AS path: I

...

224.0.0.2/32 (1 entry, 1 announced)
  *PIM Preference: 0
        Next-hop reference count: 18
        State: <Active NoReadvrt Int>
        Local AS: 69
        Age: 1:31:45
        Task: PIM Recv
        Announcement bits (2): 0-KRT 3-Resolve tree 2
        AS path: I

...

224.0.0.22/32 (1 entry, 1 announced)
  *IGMP Preference: 0
        Next-hop reference count: 18
        State: <Active NoReadvrt Int>
        Local AS: 69
        Age: 1:31:43
        Task: IGMP
        Announcement bits (2): 0-KRT 3-Resolve tree 2
        AS path: I

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

10.255.70.103/32 (1 entry, 1 announced)
  State: <FlashAll>
  *RSVP Preference: 7
        Next-hop reference count: 6
        Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
        Label-switched-path green-r1-r3
        Label operation: Push 100096
        State: <Active Int>
        Local AS: 69
        Age: 1:25:49 Metric: 2
        Task: RSVP
        Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
        AS path: I

10.255.71.238/32 (1 entry, 1 announced)
  State: <FlashAll>
  *RSVP Preference: 7
        Next-hop reference count: 6
        Next hop: via so-0/3/0.0 weight 0x1, selected
        Label-switched-path green-r1-r2
        State: <Active Int>
        Local AS: 69
        Age: 1:25:49 Metric: 1
        Task: RSVP
        Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
        AS path: I

private__inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```


47.0005.80ff.f800.0000.0108.0001.0102.5507.1052/152 (1 entry, 0 announced)

```
*Direct Preference: 0
  Next hop type: Interface
  Next-hop reference count: 1
  Next hop: via lo0.0, selected
  State: <Active Int>
  Local AS: 69
  Age: 1:31:44
  Task: IF
  AS path: I
```

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

0 (1 entry, 1 announced)

```
*MPLS Preference: 0
  Next hop type: Receive
  Next-hop reference count: 6
  State: <Active Int>
  Local AS: 69
  Age: 1:31:45 Metric: 1
  Task: MPLS
  Announcement bits (1): 0-KRT
  AS path: I
```

...

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

299776 (1 entry, 1 announced)

TSI:

KRT in-kernel 299776 /52 -> {Flood}

```
*RSVP Preference: 7
  Next hop type: Flood
  Next-hop reference count: 130
  Flood nexthop branches exceed maximum
  Address: 0x8ea65d0
```

...

800010 (1 entry, 1 announced)

```
*VPLS Preference: 7
  Next-hop reference count: 2
  Next hop: via vt-3/2/0.32769, selected
  Label operation: Pop
  State: <Active Int>
  Age: 1:29:30
  Task: Common L2 VC
  Announcement bits (1): 0-KRT
  AS path: I
```

vt-3/2/0.32769 (1 entry, 1 announced)

```
*VPLS Preference: 7
  Next-hop reference count: 2
  Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
  Label-switched-path green-r1-r3
  Label operation: Push 800012, Push 100096(top)
  Protocol next hop: 10.255.70.103
  Push 800012
  Indirect next hop: 87272e4 1048574
  State: <Active Int>
  Age: 1:29:30 Metric2: 2
  Task: Common L2 VC
  Announcement bits (2): 0-KRT 1-Common L2 VC
```

```
AS path: I
Communities: target:11111:1 Layer2-info: encaps:VPLS,
control flags:, mtu: 0

inet6.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

abcd::10:255:71:52/128 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 1
    Next hop: via lo0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:31:44
    Task: IF
    AS path: I

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 1
    Next hop: via lo0.0, selected
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:44
    Task: IF
    AS path: I

ff02::2/128 (1 entry, 1 announced)
  *PIM Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:45
    Task: PIM Recv6
    Announcement bits (1): 0-KRT
    AS path: I

ff02::d/128 (1 entry, 1 announced)
  *PIM Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:45
    Task: PIM Recv6
    Announcement bits (1): 0-KRT
    AS path: I

ff02::16/128 (1 entry, 1 announced)
  *MLD Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:43
    Task: MLD
    Announcement bits (1): 0-KRT
    AS path: I

private.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
```

```

*Direct Preference: 0
  Next hop type: Interface
  Next-hop reference count: 1
  Next hop: via lo0.16385, selected
  State: <Active NoReadvrt Int>
  Age: 1:31:44
  Task: IF
  AS path: I

green.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

10.255.70.103:1:3:1/96 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 10.255.70.103:1
    Next-hop reference count: 7
    Source: 10.255.70.103
    Protocol next hop: 10.255.70.103
    Indirect next hop: 2 no-forward
    State: <Secondary Active Int Ext>
    Local AS: 69 Peer AS: 69
    Age: 1:25:49 Metric2: 1
    AIGP 210
    Task: BGP_69.10.255.70.103+179
    Announcement bits (1): 0-green-l2vpn
    AS path: I
    Communities: target:11111:1 Layer2-info: encaps:VPLS,
    control flags:, mtu: 0
    Label-base: 800008, range: 8
    Localpref: 100
    Router ID: 10.255.70.103
    Primary Routing Table bgp.l2vpn.0

10.255.71.52:1:1:1/96 (1 entry, 1 announced)
  *L2VPN Preference: 170/-1
    Next-hop reference count: 5
    Protocol next hop: 10.255.71.52
    Indirect next hop: 0 -
    State: <Active Int Ext>
    Age: 1:31:40 Metric2: 1
    Task: green-l2vpn
    Announcement bits (1): 1-BGP.0.0.0.0+179
    AS path: I
    Communities: Layer2-info: encaps:VPLS, control flags:Site-Down,
    mtu: 0
    Label-base: 800016, range: 8, status-vector: 0x9F

10.255.71.52:1:5:1/96 (1 entry, 1 announced)
  *L2VPN Preference: 170/-101
    Next-hop reference count: 5
    Protocol next hop: 10.255.71.52
    Indirect next hop: 0 -
    State: <Active Int Ext>
    Age: 1:31:40 Metric2: 1
    Task: green-l2vpn
    Announcement bits (1): 1-BGP.0.0.0.0+179
    AS path: I
    Communities: Layer2-info: encaps:VPLS, control flags:, mtu: 0
    Label-base: 800008, range: 8, status-vector: 0x9F

...

```

```

12circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.245.255.63:CtrlWord:4:3:Local/96 (1 entry, 1 announced)
  *L2CKT Preference: 7
    Next hop: via so-1/1/2.0 weight 1, selected
    Label-switched-path my-lsp
    Label operation: Push 100000[0]
    Protocol next hop: 10.245.255.63 Indirect next hop: 86af000 296
    State: <Active Int>
    Local AS: 99
    Age: 10:21
    Task: 12 circuit
    Announcement bits (1): 0-LDP
    AS path: I
    VC Label 100000, MTU 1500, VLAN ID 512

```

show route detail (with BGP Multipath)

```
user@host> show route detail
```

```

10.1.1.8/30 (2 entries, 1 announced)
  *BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 262142
    Address: 0x901a010
    Next-hop reference count: 2
    Source: 10.1.1.2
    Next hop: 10.1.1.2 via ge-0/3/0.1, selected
    Next hop: 10.1.1.6 via ge-0/3/0.5
    State: <Active Ext>
    Local AS: 1 Peer AS: 2
    Age: 5:04:43
    Task: BGP_2.10.1.1.2+59955
    Announcement bits (1): 0-KRT
    AS path: 2 I
    Accepted Multipath
    Localpref: 100
    Router ID: 1.1.1.2
  BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 678
    Address: 0x8f97520
    Next-hop reference count: 9
    Source: 10.1.1.6
    Next hop: 10.1.1.6 via ge-0/3/0.5, selected
    State: <NotBest Ext>
    Inactive reason: Not Best in its group - Active preferred
    Local AS: 1 Peer AS: 2
    Age: 5:04:43
    Task: BGP_2.10.1.1.6+58198
    AS path: 2 I
    Accepted MultipathContrib
    Localpref: 100
    Router ID: 1.1.1.3

```

show route exact

Syntax	<code>show route exact <i>destination-prefix</i></code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Syntax (EX Series Switches)	<code>show route exact <i>destination-prefix</i></code> <code><brief detail extensive terse></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display only the routes that exactly match the specified address or range of addresses.
Options	brief detail extensive terse —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief . <i>destination-prefix</i> —Address or range of addresses. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	show route exact on page 198 show route exact detail on page 198 show route exact extensive on page 198 show route exact terse on page 198
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route exact

```
user@host> show route exact 207.17.136.0/24

inet.0: 24 destinations, 25 routes (23 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
207.17.136.0/24    *[Static/5] 2d 03:30:22
                  > to 192.168.71.254 via fxp0.0
```

show route exact detail

```
user@host> show route exact 207.17.136.0/24 detail

inet.0: 24 destinations, 25 routes (23 active, 0 holddown, 1 hidden)
Restart Complete
207.17.136.0/24 (1 entry, 1 announced)
    *Static Preference: 5
        Next-hop reference count: 29
        Next hop: 192.168.71.254 via fxp0.0, selected
        State: <Active NoReadvrt Int Ext>
        Local AS:    69
        Age: 2d 3:30:26
        Task: RT
        Announcement bits (2): 0-KRT 3-Resolve tree 2
        AS path: I
```

show route exact extensive

```
user@host> show route exact 207.17.136.0/24 extensive

inet.0: 22 destinations, 23 routes (21 active, 0 holddown, 1 hidden)
207.17.136.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 207.17.136.0/24 -> {192.168.71.254}
    *Static Preference: 5
        Next-hop reference count: 29
        Next hop: 192.168.71.254 via fxp0.0, selected
        State: <Active NoReadvrt Int Ext>
        Local AS:    69
        Age: 1:25:18
        Task: RT
        Announcement bits (2): 0-KRT 3-Resolve tree 2
        AS path: I
```

show route exact terse

```
user@host> show route exact 207.17.136.0/24 terse

inet.0: 22 destinations, 23 routes (21 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
A Destination      P Prf  Metric 1  Metric 2  Next hop      AS path
* 207.17.136.0/24  S  5                      >192.168.71.254
```

show route export

Syntax	show route export <brief detail> <instance <instance-name> routing-table-name> <logical-system (all logical-system-name)>
Syntax (EX Series Switches)	show route export <brief detail> <instance <instance-name> routing-table-name>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display policy-based route export information. Policy-based export simplifies the process of exchanging route information between routing instances.
Options	<p>none—(Same as brief.) Display standard information about policy-based export for all instances and routing tables on all systems.</p> <p>brief detail—(Optional) Display the specified level of output.</p> <p>instance <instance-name>—(Optional) Display a particular routing instance for which policy-based export is currently enabled.</p> <p>logical-system (all logical-system-name)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p>routing-table-name—(Optional) Display information about policy-based export for all routing tables whose name begins with this string (for example, inet.0 and inet6.0 are both displayed when you run the show route export inet command).</p>
Required Privilege Level	view
List of Sample Output	show route export on page 200 show route export detail on page 200 show route export instance detail on page 200
Output Fields	Table 17 on page 199 lists the output fields for the show route export command. Output fields are listed in the approximate order in which they appear.

Table 17: show route export Output Fields

Field Name	Field Description	Level of Output
Table or table-name	Name of the routing tables that either import or export routes.	All levels
Routes	Number of routes exported from this table into other tables. If a particular route is exported to different tables, the counter will only increment by one.	brief none
Export	Whether the table is currently exporting routes to other tables: Y or N (Yes or No).	brief none

Table 17: show route export Output Fields (*continued*)

Field Name	Field Description	Level of Output
Import	Tables currently importing routes from the originator table. (Not displayed for tables that are not exporting any routes.)	detail
Flags	(instance keyword only) Flags for this feature on this instance: <ul style="list-style-type: none"> config auto-policy—The policy was deduced from the configured IGP export policies. cleanup—Configuration information for this instance is no longer valid. config—The instance was explicitly configured. 	detail
Options	(instance keyword only) Configured option displays the type of routing tables the feature handles: <ul style="list-style-type: none"> unicast—Indicates <i>instance.inet.0</i>. multicast—Indicates <i>instance.inet.2</i>. unicast multicast—Indicates <i>instance.inet.0</i> and <i>instance.inet.2</i>. 	detail
Import policy	(instance keyword only) Policy that route export uses to construct the import-export matrix. Not displayed if the instance type is vrf .	detail
Instance	(instance keyword only) Name of the routing instance.	detail
Type	(instance keyword only) Type of routing instance: forwarding , non-forwarding , or vrf .	detail

Sample Output

show route export

```

user@host> show route export
Table           Export      Routes
inet.0          N           0
black.inet.0    Y           3
red.inet.0      Y           4

```

show route export detail

```

user@host> show route export detail
inet.0                      Routes:      0
black.inet.0                Routes:      3
  Import: [ inet.0 ]
red.inet.0                  Routes:      4
  Import: [ inet.0 ]

```

show route export instance detail

```

user@host> show route export instance detail
Instance: master              Type: forwarding
  Flags: <config auto-policy> Options: <unicast multicast>
  Import policy: [ (ospf-master-from-red || isis-master-from-black) ]
Instance: black               Type: non-forwarding
Instance: red                 Type: non-forwarding

```


show route extensive

Syntax	show route extensive <destination-prefix> <logical-system (all logical-system-name)>
Syntax (EX Series Switches)	show route extensive <destination-prefix>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display extensive information about the active entries in the routing tables.
Options	<p>none—Display all active entries in the routing table.</p> <p>destination-prefix—(Optional) Display active entries for the specified address or range of addresses.</p> <p>logical-system (all logical-system-name)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route extensive on page 207 show route extensive (Access Route) on page 213 show route extensive (Route Reflector) on page 213 show route extensive (FRR and LFA) on page 213 show route extensive (FRR and LFA) on page 214
Output Fields	Table 18 on page 201 describes the output fields for the show route extensive command. Output fields are listed in the approximate order in which they appear.

Table 18: show route extensive Output Fields

Field Name	Field Description
<i>routing-table-name</i>	Name of the routing table (for example, inet.0).
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.
<i>number routes</i>	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> active (routes that are active). holddown (routes that are in the pending state before being declared inactive). hidden (routes that are not used because of a routing policy).

Table 18: show route extensive Output Fields (*continued*)

Field Name	Field Description
<i>route-destination</i> (entry, announced)	<p>Route destination (for example: 10.0.0.1/24). The entry value is the number of route for this destination, and the announced value is the number of routes being announced for this destination. Sometimes the route destination is presented in another format, such as:</p> <ul style="list-style-type: none"> • MPLS-label (for example, 80001). • interface-name (for example, ge-1/0/2). • neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96). • neighbor-address—Address of the neighbor. • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord. • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport. • vc-id—Virtual circuit identifier. • source—Source of the advertisement: Local or Remote.
TSI	Protocol header information.
label stacking	<p>(Next-to-the-last-hop routing device for MPLS only) Depth of the MPLS label stack, where the label-popping operation is needed to remove one or more labels from the top of the stack. A pair of routes is displayed, because the pop operation is performed only when the stack depth is two or more labels.</p> <ul style="list-style-type: none"> • S=0 route indicates that a packet with an incoming label stack depth of two or more exits this router with one fewer label (the label-popping operation is performed). • If there is no S= information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed).
[<i>protocol, preference</i>]	<p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • -—A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route. <p>In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value and a lower Preference2 value.</p>
Level	<p>(IS-IS only). In IS-IS, a single autonomous system (AS) can be divided into smaller groups called areas. Routing between areas is organized hierarchically, allowing a domain to be administratively divided into smaller areas. This organization is accomplished by configuring Level 1 and Level 2 intermediate systems. Level 1 systems route within an area. When the destination is outside an area, they route toward a Level 2 system. Level 2 intermediate systems route between areas and toward other ASs.</p>
Route Distinguisher	IP subnet augmented with a 64-bit prefix.

Table 18: show route extensive Output Fields (*continued*)

Field Name	Field Description
Next-hop type	Type of next hop. For a description of possible values for this field, see the Output Field table in the show route detail command.
Next-hop reference count	Number of references made to the next hop.
Flood nexthop branches exceed maximum message	Indicates that the number of flood next-hop branches exceeded the system limit of 32 branches, and only a subset of the flood next-hop branches were installed in the kernel.
Source	IP address of the route source.
Next hop	Network layer address of the directly reachable neighboring system.
via	<p>Interface used to reach the next hop. If there is more than one interface available to the next hop, the name of the interface that is actually used is followed by the word Selected. This field can also contain the following information:</p> <ul style="list-style-type: none"> • Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.
Label-switched-path <i>lsp-path-name</i>	Name of the LSP used to reach the next hop.
Label operation	MPLS label and operation occurring at this routing device. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label).
Offset	Whether the metric has been increased or decreased by an offset value.
Interface	(Local only) Local interface name.
Protocol next hop	Network layer address of the remote routing device that advertised the prefix. This address is used to recursively derive a forwarding next hop.
<i>label-operation</i>	MPLS label and operation occurring at this routing device. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label).
Indirect next hops	When present, a list of nodes that are used to resolve the path to the next-hop destination, in the order that they are resolved.
State	State of the route (a route can be in more than one state). See the Output Field table in the show route detail command.

Table 18: show route extensive Output Fields (*continued*)

Field Name	Field Description
Session ID	The BFD session ID number that represents the protection using MPLS fast reroute (FRR) and loop-free alternate (LFA).
Inactive reason	<p>If the route is inactive, the reason for its current state is indicated. Typical reasons include:</p> <ul style="list-style-type: none"> • Active preferred—Currently active route was selected over this route. • Always compare MED—Path with a lower multiple exit discriminator (MED) is available. • AS path—Shorter AS path is available. • Cisco Non-deterministic MED selection—Cisco nondeterministic MED is enabled and a path with a lower MED is available. • Cluster list length—Path with a shorter cluster list length is available. • Forwarding use only—Path is only available for forwarding purposes. • IGP metric—Path through the next hop with a lower IGP metric is available. • IGP metric type—Path with a lower OSPF link-state advertisement type is available. • Interior > Exterior > Exterior via Interior—Direct, static, IGP, or EBGP path is available. • Local preference—Path with a higher local preference value is available. • Next hop address—Path with a lower metric next hop is available. • No difference—Path from a neighbor with a lower IP address is available. • Not Best in its group—Occurs when multiple peers of the same external AS advertise the same prefix and are grouped together in the selection process. When this reason is displayed, an additional reason is provided (typically one of the other reasons listed). • Number of gateways—Path with a higher number of next hops is available. • Origin—Path with a lower origin code is available. • OSPF version—Path does not support the indicated OSPF version. • RIB preference—Route from a higher-numbered routing table is available. • Route distinguisher—64-bit prefix added to IP subnets to make them unique. • Route metric or MED comparison—Route with a lower metric or MED is available. • Route preference—Route with a lower preference value is available. • Router ID—Path through a neighbor with a lower ID is available. • Unusable path—Path is not usable because of one of the following conditions: the route is damped, the route is rejected by an import policy, or the route is unresolved. • Update source—Last tiebreaker is the lowest IP address value.
Local AS	Autonomous system (AS) number of the local routing device.
Age	How long the route has been known.
AIGP	Accumulated interior gateway protocol (AIGP) BGP attribute.
Metric	Cost value of the indicated route. For routes within an AS, the cost is determined by IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.
MED-plus-IGP	Metric value for BGP path selection to which the IGP cost to the next-hop destination has been added.

Table 18: show route extensive Output Fields (*continued*)

Field Name	Field Description
TTL-Action	<p>For MPLS LSPs, state of the TTL propagation attribute. Can be enabled or disabled for all RSVP-signaled and LDP-signaled LSPs or for specific VRF routing instances.</p> <p>For sample output, see show route table.</p>
Task	Name of the protocol that has added the route.
Announcement bits	List of protocols that announce this route. n-Resolve inet indicates that the route is used for route resolution for next hops found in the routing table. n is an index used by Juniper Networks customer support only.
AS path	<p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • Recorded—The AS path is recorded by the sample process (sampled). • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device, or if AS path prepending is configured. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • [()]—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p>
AS path: I <Originator>	(For route reflected output only) Originator ID attribute set by the route reflector.
VC Label	MPLS label assigned to the Layer 2 circuit virtual connection.
MTU	Maximum transmission unit (MTU) of the Layer 2 circuit.
VLAN ID	VLAN identifier of the Layer 2 circuit.
Cluster list	(For route reflected output only) Cluster ID sent by the route reflector.
Originator ID	(For route reflected output only) Address of router that originally sent the route to the route reflector.
Prefixes bound to route	Forwarding equivalent class (FEC) bound to this route. Applicable only to routes installed by LDP.
Communities	Community path attribute for the route. See the Output Field table in the show route detail command for all possible values for this field.

Table 18: show route extensive Output Fields (*continued*)

Field Name	Field Description
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).
control flags	Control flags: none or Site Down.
mtu	Maximum transmission unit (MTU) information.
Label-Base, range	First label in a block of labels and label block size. A remote PE routing device uses this first label when sending traffic toward the advertising PE routing device.
status vector	Layer 2 VPN and VPLS network layer reachability information (NLRI).
Localpref	Local preference value included in the route.
Router ID	BGP router ID as advertised by the neighbor in the open message.
Primary Routing Table	In a routing table group, the name of the primary routing table in which the route resides.
Secondary Tables	In a routing table group, the name of one or more secondary tables in which the route resides.
Originating RIB	Name of the routing table whose active route was used to determine the forwarding next-hop entry in the resolution database. For example, in the case of inet.0 resolving through inet.0 and inet.3, this field indicates which routing table, inet.0 or inet.3, provided the best path for a particular prefix.
Node path count	Number of nodes in the path.
Forwarding nexthops	Number of forwarding next hops. The forwarding next hop is the network layer address of the directly reachable neighboring system (if applicable) and the interface used to reach it.

Sample Output

show route extensive

```

user@host> show route extensive
inet.0: 22 destinations, 23 routes (21 active, 0 holddown, 1 hidden)
10.10.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.10.0.0/16 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 29
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 1:34:06
    Task: RT
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

10.31.1.0/30 (2 entries, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 2
    Next hop: via so-0/3/0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:32:40
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I
  OSPF Preference: 10
    Next-hop reference count: 1
    Next hop: via so-0/3/0.0, selected
    State: <Int>
    Inactive reason: Route Preference
    Local AS: 69
    Age: 1:32:40 Metric: 1
    Area: 0.0.0.0
    Task: OSPF
    AS path: I

10.31.1.1/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 7
    Interface: so-0/3/0.0
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:32:43
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I

...

10.31.2.0/30 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.31.2.0/30 -> {10.31.1.6}
  *OSPF Preference: 10
    Next-hop reference count: 9
    Next hop: via so-0/3/0.0
    Next hop: 10.31.1.6 via ge-3/1/0.0, selected

```

```

        State: <Active Int>
        Local AS: 69
        Age: 1:32:19    Metric: 2
        Area: 0.0.0.0
        Task: OSPF
        Announcement bits (2): 0-KRT 3-Resolve tree 2
        AS path: I

...

224.0.0.2/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 224.0.0.2/32 -> {}
    *PIM    Preference: 0
            Next-hop reference count: 18
            State: <Active NoReadvrt Int>
            Local AS: 69
            Age: 1:34:08
            Task: PIM Recv
            Announcement bits (2): 0-KRT 3-Resolve tree 2
            AS path: I

...

224.0.0.22/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 224.0.0.22/32 -> {}
    *IGMP   Preference: 0
            Next-hop reference count: 18
            State: <Active NoReadvrt Int>
            Local AS: 69
            Age: 1:34:06
            Task: IGMP
            Announcement bits (2): 0-KRT 3-Resolve tree 2
            AS path: I

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

10.255.70.103/32 (1 entry, 1 announced)
    State: <FlashAll>
    *RSVP   Preference: 7
            Next-hop reference count: 6
            Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
            Label-switched-path green-r1-r3
            Label operation: Push 100096
            State: <Active Int>
            Local AS: 69
            Age: 1:28:12    Metric: 2
            Task: RSVP
            Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
            AS path: I

10.255.71.238/32 (1 entry, 1 announced)
    State: <FlashAll>
    *RSVP   Preference: 7
            Next-hop reference count: 6
            Next hop: via so-0/3/0.0 weight 0x1, selected
            Label-switched-path green-r1-r2
            State: <Active Int>
            Local AS: 69
            Age: 1:28:12    Metric: 1

```



```

Task: RSVP
Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
AS path: I

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

...

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

47.0005.80ff.f800.0000.0108.0001.0102.5507.1052/152 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 1
    Next hop: via lo0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:34:07
    Task: IF
    AS path: I

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

0 (1 entry, 1 announced)
TSI:
KRT in-kernel 0 /36 -> {}
  *MPLS Preference: 0
    Next hop type: Receive
    Next-hop reference count: 6
    State: <Active Int>
    Local AS: 69
    Age: 1:34:08 Metric: 1
    Task: MPLS
    Announcement bits (1): 0-KRT
    AS path: I

...

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
299776 (1 entry, 1 announced)
TSI:
KRT in-kernel 299776 /52 -> {Flood}
  *RSVP Preference: 7
    Next hop type: Flood
    Next-hop reference count: 130
    Flood nexthop branches exceed maximum
    Address: 0x8ea65d0

...

800010 (1 entry, 1 announced)

TSI:
KRT in-kernel 800010 /36 -> {vt-3/2/0.32769}
  *VPLS Preference: 7
    Next-hop reference count: 2
    Next hop: via vt-3/2/0.32769, selected
    Label operation: Pop
    State: <Active Int>
    Age: 1:31:53
    Task: Common L2 VC
    Announcement bits (1): 0-KRT

```

```

AS path: I

vt-3/2/0.32769 (1 entry, 1 announced)
TSI:
KRT in-kerne vt-3/2/0.32769.0      /16 -> {indirect(1048574)}
    *VPLS   Preference: 7
        Next-hop reference count: 2
        Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
        Label-switched-path green-r1-r3
        Label operation: Push 800012, Push 100096(top)
        Protocol next hop: 10.255.70.103
        Push 800012
        Indirect next hop: 87272e4 1048574
        State: <Active Int>
        Age: 1:31:53   Metric2: 2
        Task: Common L2 VC
        Announcement bits (2): 0-KRT 1-Common L2 VC
        AS path: I
        Communities: target:11111:1 Layer2-info: encaps:VPLS,
        control flags:, mtu: 0
        Indirect next hops: 1
            Protocol next hop: 10.255.70.103 Metric: 2
            Push 800012
            Indirect next hop: 87272e4 1048574
            Indirect path forwarding next hops: 1
                Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1
            10.255.70.103/32 Originating RIB: inet.3
                Metric: 2                               Node path count: 1
                Forwarding nexthops: 1
                    Nexthop: 10.31.1.6 via ge-3/1/0.0

inet6.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

abcd::10:255:71:52/128 (1 entry, 0 announced)
    *Direct Preference: 0
        Next hop type: Interface
        Next-hop reference count: 1
        Next hop: via lo0.0, selected
        State: <Active Int>
        Local AS: 69
        Age: 1:34:07
        Task: IF
        AS path: I

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
    *Direct Preference: 0
        Next hop type: Interface
        Next-hop reference count: 1
        Next hop: via lo0.0, selected
        State: <Active NoReadvrt Int>
        Local AS: 69
        Age: 1:34:07
        Task: IF
        AS path: I

ff02::2/128 (1 entry, 1 announced)
TSI:
KRT in-kerne ff02::2/128 -> {}
    *PIM   Preference: 0
        Next-hop reference count: 18
        State: <Active NoReadvrt Int>

```

```

Local AS: 69
Age: 1:34:08
Task: PIM Recv6
Announcement bits (1): 0-KRT
AS path: I

ff02::d/128 (1 entry, 1 announced)
TSI:
KRT in-kernel ff02::d/128 -> {}
  *PIM Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:34:08
    Task: PIM Recv6
    Announcement bits (1): 0-KRT
    AS path: I

ff02::16/128 (1 entry, 1 announced)
TSI:
KRT in-kernel ff02::16/128 -> {}
  *MLD Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:34:06
    Task: MLD
    Announcement bits (1): 0-KRT
    AS path: I

private.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 1
    Next hop: via lo0.16385, selected
    State: <Active NoReadvrt Int>
    Age: 1:34:07
    Task: IF
    AS path: I

green.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

10.255.70.103:1:3:1/96 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 10.255.70.103:1
    Next-hop reference count: 7
    Source: 10.255.70.103
    Protocol next hop: 10.255.70.103
    Indirect next hop: 2 no-forward
    State: <Secondary Active Int Ext>
    Local AS: 69 Peer AS: 69
    Age: 1:28:12 Metric2: 1
    Task: BGP_69.10.255.70.103+179
    Announcement bits (1): 0-green-l2vpn
    AS path: I
    Communities: target:11111:1 Layer2-info: encaps:VPLS,
    control flags:, mtu: 0
    Label-base: 800008, range: 8
    Localpref: 100

```

```

Router ID: 10.255.70.103
Primary Routing Table bgp.l2vpn.0

10.255.71.52:1:1:1/96 (1 entry, 1 announced)
TSI:
Page 0 idx 0 Type 1 val 8699540
    *L2VPN Preference: 170/-1
        Next-hop reference count: 5
        Protocol next hop: 10.255.71.52
        Indirect next hop: 0 -
        State: <Active Int Ext>
        Age: 1:34:03 Metric2: 1
        Task: green-l2vpn
        Announcement bits (1): 1-BGP.0.0.0+179
        AS path: I
        Communities: Layer2-info: encaps:VPLS, control flags:Site-Down,
        mtu: 0
        Label-base: 800016, range: 8, status-vector: 0x9F

10.255.71.52:1:5:1/96 (1 entry, 1 announced)
TSI:
Page 0 idx 0 Type 1 val 8699528
    *L2VPN Preference: 170/-101
        Next-hop reference count: 5
        Protocol next hop: 10.255.71.52
        Indirect next hop: 0 -
        State: <Active Int Ext>
        Age: 1:34:03 Metric2: 1
        Task: green-l2vpn
        Announcement bits (1): 1-BGP.0.0.0+179
        AS path: I
        Communities: Layer2-info: encaps:VPLS, control flags:, mtu: 0
        Label-base: 800008, range: 8, status-vector: 0x9F

...

l2circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
TSI:

10.245.255.63:CtrlWord:4:3:Local/96 (1 entry, 1 announced)
    *L2CKT Preference: 7
        Next hop: via so-1/1/2.0 weight 1, selected
        Label-switched-path my-lsp
        Label operation: Push 100000[0]
        Protocol next hop: 10.245.255.63 Indirect next hop: 86af000 296
        State: <Active Int>
        Local AS: 99
        Age: 10:21
        Task: l2 circuit
        Announcement bits (1): 0-LDP
        AS path: I
        VC Label 100000, MTU 1500, VLAN ID 512

55.0.0.0/24 (1 entry, 1 announced)
TSI:
KRT queued (pending) add
    55.0.0.0/24 -> {Push 300112}
        *BGP Preference: 170/-101
            Next hop type: Router

```

```

Address: 0x925c208
Next-hop reference count: 2
Source: 10.0.0.9
Next hop: 10.0.0.9 via ge-1/2/0.15, selected
Label operation: Push 300112
Label TTL action: prop-ttl
State: <Active Ext>
Local AS: 7019 Peer AS: 13979
Age: 1w0d 23:06:56
AIGP: 25
Task: BGP_13979.10.0.0.9+56732
Announcement bits (1): 0-KRT
AS path: 13979 7018 I
Accepted
Route Label: 300112
Localpref: 100
Router ID: 10.9.9.1

```

show route extensive (Access Route)

```

user@host> show route 13.160.0.102 extensive
inet.0: 39256 destinations, 39258 routes (39255 active, 0 holddown, 1 hidden)
13.160.0.102/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 13.160.0.102/32 -> {13.160.0.2}
OSPF area : 0.0.0.0, LSA ID : 13.160.0.102, LSA type : Extern
  *Access Preference: 13
    Next-hop reference count: 78472
    Next hop: 13.160.0.2 via fe-0/0/0.0, selected
    State: <Active Int>
  Age: 12
    Task: RPD Unix Domain Server./var/run/rpd_serv.local
    Announcement bits (2): 0-KRT 1-OSPFv2
    AS path: I

```

show route extensive (Route Reflector)

```

user@host> show route extensive
1.0.0.0/8 (1 entry, 1 announced)

TSI:
KRT in-kernel 1.0.0.0/8 -> {indirect(40)}
  *BGP Preference: 170/-101
    Source: 192.168.4.214
    Protocol next hop: 207.17.136.192 Indirect next hop: 84ac908 40
    State: <Active Int Ext>
    Local AS: 10458 Peer AS: 10458
    Age: 3:09 Metric: 0 Metric2: 0
    Task: BGP_10458.192.168.4.214+1033
    Announcement bits (2): 0-KRT 4-Resolve inet.0
    AS path: 3944 7777 I <Originator>
    Cluster list: 1.1.1.1
    Originator ID: 10.255.245.88
    Communities: 7777:7777
    Localpref: 100
    Router ID: 4.4.4.4
    Indirect next hops: 1
      Protocol next hop: 207.17.136.192 Metric: 0
      Indirect next hop: 84ac908 40
      Indirect path forwarding next hops: 0
      Next hop type: Discard

```

```

user@host> show route 20:31:2:0 extensive

```

**show route extensive
(FRR and LFA)**

```

inet.0: 46 destinations, 49 routes (45 active, 0 holddown, 1 hidden)
20.31.2.0/24 (2 entries, 1 announced)
  State: FlashAll
TSI:
KRT in-kernel 20.31.2.0/24 -> {Push 299776, Push 299792}
  *RSVP Preference: 7/1
    Next hop type: Router, Next hop index: 1048574
    Address: 0xbbbc010
    Next-hop reference count: 5
    Next hop: 10.31.1.2 via ge-2/1/8.0 weight 0x1, selected
    Label-switched-path europa-d-to-europa-e
    Label operation: Push 299776
    Label TTL action: prop-ttl
    Session Id: 0x201
    Next hop: 10.31.2.2 via ge-2/1/4.0 weight 0x4001
    Label-switched-path europa-d-to-europa-e
    Label operation: Push 299792
    Label TTL action: prop-ttl
    Session Id: 0x202
    State: Active Int
    Local AS: 100
    Age: 5:31 Metric: 2
    Task: RSVP
    Announcement bits (1): 0-KRT
    AS path: I
  OSPF Preference: 10
    Next hop type: Router, Next hop index: 615
    Address: 0xb9d78c4
    Next-hop reference count: 7
    Next hop: 10.31.1.2 via ge-2/1/8.0, selected
    Session Id: 0x201
    State: Int
    Inactive reason: Route Preference
    Local AS: 100
    Age: 5:35 Metric: 3
    Area: 0.0.0.0
    Task: OSPF
    AS path: I

```

**show route extensive
(FRR and LFA)**

```

user@host> show route 20.31.2.0 extensive
inet.0: 46 destinations, 49 routes (45 active, 0 holddown, 1 hidden)
20.31.2.0/24 (2 entries, 1 announced)
  State: FlashAll
TSI:
KRT in-kernel 20.31.2.0/24 -> {Push 299776, Push 299792}
  *RSVP Preference: 7/1
    Next hop type: Router, Next hop index: 1048574
    Address: 0xbbbc010
    Next-hop reference count: 5
    Next hop: 10.31.1.2 via ge-2/1/8.0 weight 0x1, selected
    Label-switched-path europa-d-to-europa-e
    Label operation: Push 299776
    Label TTL action: prop-ttl
    Session Id: 0x201
    Next hop: 10.31.2.2 via ge-2/1/4.0 weight 0x4001
    Label-switched-path europa-d-to-europa-e
    Label operation: Push 299792
    Label TTL action: prop-ttl
    Session Id: 0x202
    State: Active Int
    Local AS: 100

```

```
Age: 5:31 Metric: 2
Task: RSVP
Announcement bits (1): 0-KRT
AS path: I
OSPF Preference: 10
Next hop type: Router, Next hop index: 615
Address: 0xb9d78c4
Next-hop reference count: 7
Next hop: 10.31.1.2 via ge-2/1/8.0, selected
Session Id: 0x201
State: Int
Inactive reason: Route Preference
Local AS: 100
Age: 5:35 Metric: 3
Area: 0.0.0.0
Task: OSPF
AS path: I
```

show route forwarding-table

Syntax	<pre>show route forwarding-table <detail extensive summary> <all> <ccc interface-name> <destination destination-prefix> <family family matching matching> <interface-name interface-name> <label name> <matching matching> <multicast> <table (default logical-system-name/routing-instance-name routing-instance-name)> <vlan (all vlan-name)> <vpn vpn></pre>
Syntax (MX Series Routers)	<pre>show route forwarding-table <detail extensive summary> <all> <bridge-domain (all domain-name)> <ccc interface-name> <destination destination-prefix> <family family matching matching> <interface-name interface-name> <label name> <learning-vlan-id learning-vlan-id> <matching matching> <multicast> <table (default logical-system-name/routing-instance-name routing-instance-name)> <vlan (all vlan-name)> <vpn vpn></pre>
Syntax (Routing Matrix)	<pre>show route forwarding-table <detail extensive summary> <all> <ccc interface-name> <destination destination-prefix> <family family matching matching> <interface-name interface-name> <matching matching> <label name> <lcc number> <multicast> <table routing-instance-name> <vpn vpn></pre>
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Option bridge-domain introduced in Junos OS Release 7.5</p> <p>Option learning-vlan-id introduced in Junos OS Release 8.4</p> <p>Options all and vlan introduced in Junos OS Release 9.6.</p> <p>Command introduced in Junos OS Release 11.3 for the QFX Series.</p>

Description Display the Routing Engine's forwarding table, including the network-layer prefixes and their next hops. This command is used to help verify that the routing protocol process has relayed the correction information to the forwarding table. The Routing Engine constructs and maintains one or more routing tables. From the routing tables, the Routing Engine derives a table of active routes, called the forwarding table.



NOTE: The Routing Engine copies the forwarding table to the Packet Forwarding Engine, the part of the router that is responsible for forwarding packets. To display the entries in the Packet Forwarding Engine's forwarding table, use the **show pfe route** command.

Options **none**—Display the routes in the forwarding tables. By default, the **show route forwarding-table** command does not display information about private, or internal, forwarding tables.

detail | extensive | summary—(Optional) Display the specified level of output.

all—(Optional) Display routing table entries for all forwarding tables, including private, or internal, tables.

bridge-domain (all | bridge-domain-name)—(MX Series routers only) (Optional) Display route entries for all bridge domains or the specified bridge domain.

ccc interface-name—(Optional) Display route entries for the specified circuit cross-connect interface.

destination destination-prefix—(Optional) Destination prefix.

family family—(Optional) Display routing table entries for the specified family: **fibre-channel**, **fmembers**, **inet**, **inet6**, **iso**, **mpls**, **tnp**, **unix**, **vpls**, or **vlan-classification**.

interface-name interface-name—(Optional) Display routing table entries for the specified interface.

label name—(Optional) Display route entries for the specified label.

lcc number—(Routing Matrix only) (Optional) On a routing matrix composed of a TX Matrix Plus router and T640 routers configured in the routing matrix, display information for the specified T640 router (or line-card chassis) connected to the TX Matrix router. On a routing matrix composed of the TX Matrix Plus router and T1600 routers configured in the routing matrix, display information for the specified T1600 router (or line-card chassis) connected to the TX Matrix Plus router. Replace **number** with a value from 0 through 3.

learning-vlan-id learning-vlan-id—(MX Series routers only) (Optional) Display learned information for all VLANs or for the specified VLAN.

matching matching—(Optional) Display routing table entries matching the specified prefix or prefix length.

multicast—(Optional) Display routing table entries for multicast routes.

table (**default** | *logical-system-name/routing-instance-name* |

routing-instance-name)—(Optional) Display route entries for all the routing tables in the main routing instance or for the specified routing instance. If your device supports logical systems, you can also display route entries for the specified logical system and routing instance. To view the routing instances on your device, use the [show route instance](#) command.

vlan (**all** | *vlan-name*)—(Optional) Display information for all VLANs or for the specified VLAN.

vpn *vpn*—(Optional) Display routing table entries for a specified VPN.

Required Privilege Level view

List of Sample Output [show route forwarding-table on page 221](#)
[show route forwarding-table detail on page 221](#)
[show route forwarding-table destination extensive \(Weights and Balances\) on page 222](#)
[show route forwarding-table extensive on page 223](#)
[show route forwarding-table extensive \(RPF\) on page 224](#)
[show route forwarding-table family mpls on page 226](#)
[show route forwarding-table family vpls on page 226](#)
[show route forwarding-table family vpls extensive on page 226](#)
[show route forwarding-table table default on page 227](#)
[show route forwarding-table table](#)
[logical-system-name/routing-instance-name on page 228](#)
[show route forwarding-table vpn on page 229](#)

Output Fields [Table 19 on page 218](#) lists the output fields for the **show route forwarding-table** command. Output fields are listed in the approximate order in which they appear. Field names might be abbreviated (as shown in parentheses) when no level of output is specified, or when the **detail** keyword is used instead of the **extensive** keyword.

Table 19: show route forwarding-table Output Fields

Field Name	Field Description	Level of Output
Logical system	Name of the logical system. This field is displayed if you specify the table <i>logical-system-name/routing-instance-name</i> option on a device that is configured for and supports logical systems.	All levels
Routing table	Name of the routing table (for example, inet, inet6, mpls).	All levels
Address family	Address family (for example, IP, IPv6, ISO, MPLS, and VPLS).	All levels
Destination	Destination of the route.	detail extensive

Table 19: show route forwarding-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Route Type (Type)	<p>How the route was placed into the forwarding table. When the detail keyword is used, the route type might be abbreviated (as shown in parentheses):</p> <ul style="list-style-type: none"> • cloned (clon)—(TCP or multicast only) Cloned route. • destination (dest)—Remote addresses directly reachable through an interface. • destination down (iddn)—Destination route for which the interface is unreachable. • interface cloned (ifcl)—Cloned route for which the interface is unreachable. • route down (ifdn)—Interface route for which the interface is unreachable. • ignore (ignr)—Ignore this route. • interface (intf)—Installed as a result of configuring an interface. • permanent (perm)—Routes installed by the kernel when the routing table is initialized. • user—Routes installed by the routing protocol process or as a result of the configuration. 	All levels
Route Reference (RtRef)	Number of routes to reference.	detail extensive
Flags	<p>Route type flags:</p> <ul style="list-style-type: none"> • none—No flags are enabled. • accounting—Route has accounting enabled. • cached—Cache route. • incoming-iface interface-number—Check against incoming interface. • prefix load balance—Load balancing is enabled for this prefix. • rt nh decoupled—Route has been decoupled from the next hop to the destination. • sent to PFE—Route has been sent to the Packet Forwarding Engine. • static—Static route. 	extensive
Next hop	IP address of the next hop to the destination.	detail extensive

Table 19: show route forwarding-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Next hop Type (Type)	<p>Next-hop type. When the detail keyword is used, the next-hop type might be abbreviated (as indicated in parentheses):</p> <ul style="list-style-type: none"> • broadcast (bcst)—Broadcast. • deny—Deny. • discard (dscd)—Discard. • hold—Next hop is waiting to be resolved into a unicast or multicast type. • indexed (idxd)—Indexed next hop. • indirect (indr)—Indirect next hop. • local (locl)—Local address on an interface. • routed multicast (mcrst)—Regular multicast next hop. • multicast (mcst)—Wire multicast next hop (limited to the LAN). • multicast discard (mdsc)—Multicast discard. • multicast group (mgrp)—Multicast group member. • receive (rcv)—Receive. • reject (rjct)—Discard. An ICMP unreachable message was sent. • resolve (rslv)—Resolving the next hop. • unicast (ucst)—Unicast. • unilist (ulst)—List of unicast next hops. A packet sent to this next hop goes to any next hop in the list. 	detail extensive
Index	Software index of the next hop that is used to route the traffic for a given prefix.	detail extensive none
Route interface-index	Logical interface index from which the route is learned. For example, for interface routes, this is the logical interface index of the route itself. For static routes, this field is zero. For routes learned through routing protocols, this is the logical interface index from which the route is learned.	extensive
Reference (NhRef)	Number of routes that refer to this next hop.	detail extensive none
Next-hop interface (Netif)	Interface used to reach the next hop.	detail extensive none
Weight	Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible (see the Balance field description).	extensive
Balance	Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a router is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.	extensive
RPF interface	List of interfaces from which the prefix can be accepted. Reverse path forwarding (RPF) information is displayed only when rpf-check is configured on the interface.	extensive

Sample Output

```
show route
forwarding-table
```

```
user@host> show route forwarding-table
```

```
Routing table: default.inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	46	4	
0.0.0.0/32	perm	0		dscd	44	1	
1.1.1.0/24	ifdn	0		rslv	608	1	ge-2/0/1.0
1.1.1.0/32	iddn	0	1.1.1.0	recv	606	1	ge-2/0/1.0
1.1.1.1/32	user	0		rjct	46	4	
1.1.1.1/32	intf	0	1.1.1.1	loc1	607	2	
1.1.1.1/32	iddn	0	1.1.1.1	loc1	607	2	
1.1.1.255/32	iddn	0	ff:ff:ff:ff:ff:ff	bcst	605	1	ge-2/0/1.0
10.0.0.0/24	intf	0		rslv	616	1	ge-2/0/0.0
10.0.0.0/32	dest	0	10.0.0.0	recv	614	1	ge-2/0/0.0
10.0.0.1/32	intf	0	10.0.0.1	loc1	615	2	
10.0.0.1/32	dest	0	10.0.0.1	loc1	615	2	
10.0.0.255/32	dest	0	10.0.0.255	bcst	613	1	ge-2/0/0.0
10.1.1.0/24	ifdn	0		rslv	612	1	ge-2/0/1.0
10.1.1.0/32	iddn	0	10.1.1.0	recv	610	1	ge-2/0/1.0
10.1.1.1/32	user	0		rjct	46	4	
10.1.1.1/32	intf	0	10.1.1.1	loc1	611	2	
10.1.1.1/32	iddn	0	10.1.1.1	loc1	611	2	
10.1.1.255/32	iddn	0	ff:ff:ff:ff:ff:ff	bcst	609	1	ge-2/0/1.0
10.209.0.0/16	user	0	10.209.63.254	ucst	419	20	fxp0.0
10.209.0.0/16	user	1	0:12:1e:ca:98:0	ucst	419	20	fxp0.0
10.209.0.0/18	intf	0		rslv	418	1	fxp0.0
10.209.0.0/32	dest	0	10.209.0.0	recv	416	1	fxp0.0
10.209.2.131/32	intf	0	10.209.2.131	loc1	417	2	
10.209.2.131/32	dest	0	10.209.2.131	loc1	417	2	
10.209.17.55/32	dest	0	0:30:48:5b:78:d2	ucst	435	1	fxp0.0
10.209.63.42/32	dest	0	0:23:7d:58:92:ca	ucst	434	1	fxp0.0
10.209.63.254/32	dest	0	0:12:1e:ca:98:0	ucst	419	20	fxp0.0
10.209.63.255/32	dest	0	10.209.63.255	bcst	415	1	fxp0.0
10.227.0.0/16	user	0	10.209.63.254	ucst	419	20	fxp0.0

```
...
```

```
Routing table: iso
```

```
ISO:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	27	1	
47.0005.80ff.f800.0000.0108.0003.0102.5524.5220.00	intf	0	loc1 28 1				

```
Routing table: inet6
```

```
Internet6:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	6	1	
ff00::/8	perm	0		mdsc	4	1	
ff02::1/128	perm	0	ff02::1	mcst	3	1	

```
Routing table: ccc
```

```
MPLS:
```

Interface.Label	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	16	1	
100004(top)fe-0/0/1.0							

show route forwarding-table detail

```
user@host> show route forwarding-table detail
```

```
Routing table: inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	user	2	0:90:69:8e:b1:1b	ucst	132	4	fxp0.0
default	perm	0		rjct	14	1	
10.1.1.0/24	intf	0	ff.3.0.21	ucst	322	1	so-5/3/0.0
10.1.1.0/32	dest	0	10.1.1.0	recv	324	1	so-5/3/0.0
10.1.1.1/32	intf	0	10.1.1.1	loc1	321	1	
10.1.1.255/32	dest	0	10.1.1.255	bcst	323	1	so-5/3/0.0
10.21.21.0/24	intf	0	ff.3.0.21	ucst	326	1	so-5/3/0.0
10.21.21.0/32	dest	0	10.21.21.0	recv	328	1	so-5/3/0.0
10.21.21.1/32	intf	0	10.21.21.1	loc1	325	1	
10.21.21.255/32	dest	0	10.21.21.255	bcst	327	1	so-5/3/0.0
127.0.0.1/32	intf	0	127.0.0.1	loc1	320	1	
172.17.28.19/32	clon	1	192.168.4.254	ucst	132	4	fxp0.0
172.17.28.44/32	clon	1	192.168.4.254	ucst	132	4	fxp0.0

```
...
```

```
Routing table: private1__inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	46	1	
10.0.0.0/8	intf	0		rslv	136	1	fxp1.0
10.0.0.0/32	dest	0	10.0.0.0	recv	134	1	fxp1.0
10.0.0.4/32	intf	0	10.0.0.4	loc1	135	2	
10.0.0.4/32	dest	0	10.0.0.4	loc1	135	2	

```
...
```

```
Routing table: iso
```

```
ISO:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	38	1	

```
Routing table: inet6
```

```
Internet6:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	22	1	
ff00::/8	perm	0		mdsc	21	1	
ff02::1/128	perm	0	ff02::1	mcst	17	1	

```
...
```

```
Routing table: mpls
```

```
MPLS:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	28	1	

show route forwarding-table destination extensive

```
user@host> show route forwarding-table destination 3.4.2.1 extensive
```

```
Routing table: inet [Index 0]
```

```
Internet:
```

(Weights and Balances)

```

Destination: 3.4.2.1/32
Route type: user
Route reference: 0
Flags: sent to PFE
Next-hop type: unicast
Nexthop: 4.4.4.4
Next-hop type: unicast
Next-hop interface: so-1/1/0.0
Nexthop: 145.12.1.2
Next-hop type: unicast
Next-hop interface: so-0/1/2.0
Route interface-index: 0
Index: 262143 Reference: 1
Index: 335 Reference: 2
Weight: 22 Balance: 3
Index: 337 Reference: 2
Weight: 33 Balance: 33

```

show route forwarding-table extensive

```

user@host> show route forwarding-table extensive
Routing table: inet [Index 0]
Internet:

Destination: default
Route type: user
Route reference: 2
Flags: sent to PFE
Nexthop: 0:90:69:8e:b1:1b
Next-hop type: unicast
Next-hop interface: fxp0.0
Route interface-index: 0
Index: 132 Reference: 4

Destination: default
Route type: permanent
Route reference: 0
Flags: none
Next-hop type: reject
Route interface-index: 0
Index: 14 Reference: 1

Destination: 127.0.0.1/32
Route type: interface
Route reference: 0
Flags: sent to PFE
Nexthop: 127.0.0.1
Next-hop type: local
Route interface-index: 0
Index: 320 Reference: 1

...

Routing table: private1__inet [Index 1]
Internet:

Destination: default
Route type: permanent
Route reference: 0
Flags: sent to PFE
Next-hop type: reject
Route interface-index: 0
Index: 46 Reference: 1

Destination: 10.0.0.0/8
Route type: interface
Route reference: 0
Flags: sent to PFE
Next-hop type: resolve
Next-hop interface: fxp1.0
Route interface-index: 3
Index: 136 Reference: 1

...

Routing table: iso [Index 0]
ISO:

Destination: default

```

```

Route type: permanent
Route reference: 0
Flags: sent to PFE
Next-hop type: reject
Route interface-index: 0
Index: 38      Reference: 1

Routing table: inet6 [Index 0]
Internet6:

Destination: default
Route type: permanent
Route reference: 0
Flags: sent to PFE
Next-hop type: reject
Route interface-index: 0
Index: 22      Reference: 1

Destination: ff00::/8
Route type: permanent
Route reference: 0
Flags: sent to PFE
Next-hop type: multicast discard
Route interface-index: 0
Index: 21      Reference: 1

...

Routing table: private1__inet6 [Index 1]
Internet6:

Destination: default
Route type: permanent
Route reference: 0
Flags: sent to PFE
Next-hop type: reject
Route interface-index: 0
Index: 54      Reference: 1

Destination: fe80::2a0:a5ff:fe3d:375/128
Route type: interface
Route reference: 0
Flags: sent to PFE
Nexthop: fe80::2a0:a5ff:fe3d:375
Next-hop type: local
Route interface-index: 0
Index: 75      Reference: 1

...

```

show route forwarding-table extensive (RPF)

The next example is based on the following configuration, which enables an RPF check on all routes that are learned from this interface, including the interface route:

```

so-1/1/0 {
  unit 0 {
    family inet {
      rpf-check;
      address 15.95.1.2/30;
    }
  }
}

```

```

user@host> show route forwarding-table extensive
Routing table: inet [Index 0]
Internet:
...
...
Destination: 15.95.1.3/32
Route type: destination
Route reference: 0
Route interface-index: 67

```


Flags: sent to PFE		
Nexthop: 15.95.1.3		
Next-hop type: broadcast	Index: 328	Reference: 1
Next-hop interface: so-1/1/0.0		
RPF interface: so-1/1/0.0		

show route forwarding-table family mpls

```

user@host> show route forwarding-table family mpls
Routing table: mpls
MPLS:
Destination      Type RtRef Next hop      Type Index NhRef Netif
default          perm  0
0                user  0
1                user  0
2                user  0
100000           user  0 10.31.1.6      swap 100001      fe-1/1/0.0
800002           user  0                Pop                vt-0/3/0.32770

vt-0/3/0.32770 (VPLS)
                  user  0                indr  351      4
                  Push 800000, Push 100002(top)

so-0/0/0.0

```

show route forwarding-table family vpls

```

user@host> show route forwarding-table family vpls
Routing table: green.vpls
VPLS:
Destination      Type RtRef Next hop      Type Index NhRef Netif
default          dynm  0
default          perm  0
fe-0/1/0.0       dynm  0
00:90:69:0c:20:1f/48      <<<<<Remote CE

                  dynm  0                indr  351      4
                  Push 800000, Push 100002(top)

so-0/0/0.0
00:90:69:85:b0:1f/48      <<<<<Local CE

                  dynm  0                ucst  354      2 fe-0/1/0.0

```

show route forwarding-table family vpls extensive

```

user@host> show route forwarding-table family vpls extensive
Routing table: green.vpls [Index 2]
VPLS:

Destination: default
Route type: dynamic
Route reference: 0
Flags: sent to PFE
Next-hop type: flood
Next-hop type: unicast
Next-hop interface: fe-0/1/3.0
Next-hop type: unicast
Next-hop interface: fe-0/1/2.0
Route interface-index: 72
Index: 289 Reference: 1
Index: 291 Reference: 3
Index: 290 Reference: 3

Destination: default
Route type: permanent
Route reference: 0
Flags: none
Next-hop type: discard
Route interface-index: 0
Index: 341 Reference: 1

Destination: fe-0/1/2.0
Route type: dynamic
Route reference: 0
Flags: sent to PFE
Next-hop type: flood
Next-hop type: indirect
Next-hop type: Push 800016
Route interface-index: 69
Index: 293 Reference: 1
Index: 363 Reference: 4

```

```

Next-hop interface: at-1/0/1.0
Next-hop type: indirect          Index: 301      Reference: 5
Next hop: 10.31.3.2
Next-hop type: Push 800000
Next-hop interface: fe-0/1/1.0
Next-hop type: unicast          Index: 291      Reference: 3
Next-hop interface: fe-0/1/3.0

Destination: fe-0/1/3.0
Route type: dynamic
Route reference: 0               Route interface-index: 70
Flags: sent to PFE
Next-hop type: flood            Index: 292      Reference: 1
Next-hop type: indirect        Index: 363      Reference: 4
Next-hop type: Push 800016
Next-hop interface: at-1/0/1.0
Next-hop type: indirect        Index: 301      Reference: 5
Next hop: 10.31.3.2
Next-hop type: Push 800000
Next-hop interface: fe-0/1/1.0
Next-hop type: unicast          Index: 290      Reference: 3
Next-hop interface: fe-0/1/2.0

Destination: 10:00:00:01:01:01/48
Route type: dynamic
Route reference: 0               Route interface-index: 70
Flags: sent to PFE, prefix load balance
Next-hop type: unicast          Index: 291      Reference: 3
Next-hop interface: fe-0/1/3.0
Route used as destination:
  Packet count:      6640   Byte count:      675786
Route used as source
  Packet count:      6894   Byte count:      696424

Destination: 10:00:00:01:01:01:04/48
Route type: dynamic
Route reference: 0               Route interface-index: 69
Flags: sent to PFE, prefix load balance
Next-hop type: unicast          Index: 290      Reference: 3
Next-hop interface: fe-0/1/2.0
Route used as destination:
  Packet count:      96     Byte count:      8079
Route used as source:
  Packet count:      296    Byte count:      24955

Destination: 10:00:00:01:03:05/48
Route type: dynamic
Route reference: 0               Route interface-index: 74
Flags: sent to PFE, prefix load balance
Next-hop type: indirect        Index: 301      Reference: 5
Next hop: 10.31.3.2
Next-hop type: Push 800000
Next-hop interface: fe-0/1/1.0

```

show route forwarding-table table default

```

user@host> show route forwarding-table table default
Routing table: default.inet

```

```

Internet:
Destination      Type RtRef Next hop      Type Index NhRef Netif
default          perm  0
0.0.0.0/32       perm  0
10.0.60.0/30     user  0 10.0.60.13  ucst  713  5 fe-0/1/3.0

```

```

10.0.60.12/30    intf    0          rslv    688    1 fe-0/1/3.0
10.0.60.12/32    dest    0 10.0.60.12    recv    686    1 fe-0/1/3.0
10.0.60.13/32    dest    0 0:5:85:8b:bc:22    ucst    713    5 fe-0/1/3.0
10.0.60.14/32    intf    0 10.0.60.14    locl    687    2
10.0.60.14/32    dest    0 10.0.60.14    locl    687    2
10.0.60.15/32    dest    0 10.0.60.15    bcst    685    1 fe-0/1/3.0
10.0.67.12/30    user    0 10.0.60.13    ucst    713    5 fe-0/1/3.0
10.0.80.0/30     ifdn    0 ff.3.0.21     ucst    676    1 so-0/0/1.0
10.0.80.0/32     dest    0 10.0.80.0     recv    678    1 so-0/0/1.0
10.0.80.2/32     user    0          rjct    36     2
10.0.80.2/32     intf    0 10.0.80.2     locl    675    1
10.0.80.3/32     dest    0 10.0.80.3     bcst    677    1 so-0/0/1.0
10.0.90.12/30    intf    0          rslv    684    1 fe-0/1/0.0
10.0.90.12/32    dest    0 10.0.90.12    recv    682    1 fe-0/1/0.0
10.0.90.14/32    intf    0 10.0.90.14    locl    683    2
10.0.90.14/32    dest    0 10.0.90.14    locl    683    2
10.0.90.15/32    dest    0 10.0.90.15    bcst    681    1 fe-0/1/0.0
10.5.0.0/16      user    0 192.168.187.126    ucst    324    15 fxp0.0
10.10.0.0/16     user    0 192.168.187.126    ucst    324    15 fxp0.0
10.13.10.0/23    user    0 192.168.187.126    ucst    324    15 fxp0.0
10.84.0.0/16     user    0 192.168.187.126    ucst    324    15 fxp0.0
10.150.0.0/16    user    0 192.168.187.126    ucst    324    15 fxp0.0
10.157.64.0/19   user    0 192.168.187.126    ucst    324    15 fxp0.0
10.209.0.0/16    user    0 192.168.187.126    ucst    324    15 fxp0.0

```

...

Routing table: default.iso

ISO:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	60	1	

Routing table: default.inet6

Internet6:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	44	1	
::/128	perm	0		dscd	42	1	
ff00::/8	perm	0		mdsc	43	1	
ff02::1/128	perm	0	ff02::1	mcst	39	1	

Routing table: default.mpls

MPLS:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		dscd	50	1	

show route
forwarding-table table
logical-system-name

user@host> show route forwarding-table table R4/vpn-red

Logical system: R4

Routing table: vpn-red.inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	563	1	
0.0.0.0/32	perm	0		dscd	561	2	
1.0.0.1/32	user	0		dscd	561	2	
2.0.2.0/24	intf	0		rslv	771	1	ge-1/2/0.3
2.0.2.0/32	dest	0	2.0.2.0	recv	769	1	ge-1/2/0.3
2.0.2.1/32	intf	0	2.0.2.1	locl	770	2	
2.0.2.1/32	dest	0	2.0.2.1	locl	770	2	
2.0.2.2/32	dest	0	0.4.80.3.0.1b.c0.d5.e4.bd.0.1b.c0.d5.e4.bc.8.0	ucst	789	1	ge-1/2/0.3
2.0.2.255/32	dest	0	2.0.2.255	bcst	768	1	ge-1/2/0.3
224.0.0.0/4	perm	1		mdsc	562	1	

```

224.0.0.1/32      perm    0 224.0.0.1      mcst   558    1
255.255.255.255/32 perm    0                      bcst   559    1

```

Logical system: R4

Routing table: vpn-red.iso

ISO:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	608	1	

Logical system: R4

Routing table: vpn-red.inet6

Internet6:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	708	1	
::/128	perm	0		dscd	706	1	
ff00::/8	perm	0		mdsc	707	1	
ff02::1/128	perm	0	ff02::1	mcst	704	1	

Logical system: R4

Routing table: vpn-red.mpls

MPLS:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		dscd	638		

**show route
forwarding-table vpn**

user@host> **show route forwarding-table vpn VPN-A**

Routing table:: VPN-A.inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	4	4	
10.39.10.20/30	intf	0	ff.3.0.21	ucst	40	1	
so-0/0/0.0							
10.39.10.21/32	intf	0	10.39.10.21	loc1	36	1	
10.255.14.172/32	user	0		ucst	69	2	
so-0/0/0.0							
10.255.14.175/32	user	0		indr	81	3	
				Push	100004	Push	
100004(top) so-1/0/0.0							
224.0.0.0/4	perm	2		mdsc	5	3	
224.0.0.1/32	perm	0	224.0.0.1	mcst	1	8	
224.0.0.5/32	user	1	224.0.0.5	mcst	1	8	
255.255.255.255/32	perm	0		bcst	2	3	

show route hidden

Syntax	<code>show route hidden</code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Release Information	Command introduced before Junos OS Release 7.4.
Description	Display only hidden route information. A hidden route is unusable, even if it is the best path.
Options	brief detail extensive terse —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief . logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	show route hidden on page 231 show route hidden detail on page 231 show route hidden extensive on page 232 show route hidden terse on page 232
Output Fields	For information about output fields, see the output field table for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route hidden

```

user@host> show route hidden
inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
127.0.0.1/32      [Direct/0] 04:26:38
                  > via lo0.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
10.5.5.5/32      [BGP/170] 03:44:10, localpref 100, from 10.4.4.4
                  AS path: 100 I
                  Unusable
10.12.1.0/24     [BGP/170] 03:44:10, localpref 100, from 10.4.4.4
                  AS path: 100 I
                  Unusable
10.12.80.4/30    [BGP/170] 03:44:10, localpref 100, from 10.4.4.4
                  AS path: I
                  Unusable
...

```

show route hidden detail

```

user@host> show route hidden detail

inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
Restart Complete
127.0.0.1/32 (1 entry, 0 announced)
    Direct Preference: 0
        Next hop type: Interface
        Next-hop reference count: 1
        Next hop: via lo0.0, selected
        State: <Hidden Martian Int>
        Local AS: 1
        Age: 4:27:37
        Task: IF
        AS path: I

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
Restart Complete
10.5.5.5/32 (1 entry, 0 announced)
    BGP Preference: 170/-101
        Route Distinguisher: 10.4.4.4:4
        Next hop type: Unusable
        Next-hop reference count: 6
        State: <Secondary Hidden Int Ext>
        Local AS: 1 Peer AS: 1
        Age: 3:45:09
        Task: BGP_1.10.4.4.4+2493
        AS path: 100 I
        Communities: target:1:999
        VPN Label: 100064
        Localpref: 100
        Router ID: 10.4.4.4

```

Primary Routing Table bgp.13vpn.0

...

show route hidden extensive

The output for the **show route hidden extensive** command is identical to that of the **show route hidden detail** command. For sample output, see [show route hidden detail on page 231](#).

show route hidden terse

```
user@host> show route hidden terse
```

```
inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
```

```
Restart Complete
```

```
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
127.0.0.1/32	D 0			>100.0	

```
private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
```

```
red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
```

```
Restart Complete
```

```
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
10.5.5.5/32	B 170	100		Unusable	100 I
10.12.1.0/24	B 170	100		Unusable	100 I
10.12.80.4/30	B 170	100		Unusable	I

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
bgp.13vpn.0: 3 destinations, 3 routes (0 active, 0 holddown, 3 hidden)
```

```
Restart Complete
```

```
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
10.4.4.4:4:10.5.5.5/32	B 170	100		Unusable	100 I
10.4.4.4:4:10.12.1.0/24	B 170	100		Unusable	100 I
10.4.4.4:4:10.12.80.4/30	B 170	100		Unusable	I

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```


show route inactive-path

Syntax	show route inactive-path <brief detail extensive terse> <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches)	show route inactive-path <brief detail extensive terse>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display routes for destinations that have no active route. An inactive route is a route that was not selected as the best path.
Options	<p>none—Display all inactive routes.</p> <p>brief detail extensive terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route inactive-path on page 234 show route inactive-path detail on page 234 show route inactive-path extensive on page 235 show route inactive-path terse on page 235
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route inactive-path

```
user@host> show route inactive-path

inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.12.100.12/30      [OSPF/10] 03:57:28, metric 1
> via so-0/3/0.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/8          [Direct/0] 04:39:56
> via fxp1.0

red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.12.80.0/30       [BGP/170] 04:38:17, localpref 100
AS path: 100 I
> to 10.12.80.1 via ge-6/3/2.0

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

bgp.l3vpn.0: 3 destinations, 3 routes (0 active, 0 holddown, 3 hidden)
Restart Complete

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route inactive-path detail

```
user@host> show route inactive-path detail

inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
Restart Complete

10.12.100.12/30 (2 entries, 1 announced)
  OSPF   Preference: 10
        Next-hop reference count: 1
        Next hop: via so-0/3/0.0, selected
        State: <Int>
        Inactive reason: Route Preference
        Local AS:      1
        Age: 3:58:24   Metric: 1
        Area: 0.0.0.0
        Task: OSPF
        AS path: I

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

10.0.0.0/8 (2 entries, 0 announced)
```

```

Direct Preference: 0
  Next hop type: Interface
  Next-hop reference count: 1
  Next hop: via fxp1.0, selected
  State: <NotBest Int>
  Inactive reason: No difference
  Age: 4:40:52
  Task: IF
  AS path: I

red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
Restart Complete

10.12.80.0/30 (2 entries, 1 announced)
  BGP Preference: 170/-101
  Next-hop reference count: 6
  Source: 10.12.80.1
  Next hop: 10.12.80.1 via ge-6/3/2.0, selected
  State: <Ext>
  Inactive reason: Route Preference
  Peer AS: 100
  Age: 4:39:13
  Task: BGP_100.10.12.80.1+179
  AS path: 100 I
  Localpref: 100
  Router ID: 10.0.0.0

```

**show route
inactive-path
extensive**

The output for the **show route inactive-path extensive** command is identical to that of the **show route inactive-path detail** command. For sample output, see [show route inactive-path detail on page 234](#).

**show route
inactive-path terse**

```

user@host> show route inactive-path terse

inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1  Metric 2  Next hop      AS path
  10.12.100.12/30   0 10      1          >so-0/3/0.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1  Metric 2  Next hop      AS path
  10.0.0.0/8        D 0          >fxp1.0

red.inet.0: 6 destinations, 8 routes (4 active, 0 holddown, 3 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1  Metric 2  Next hop      AS path
  10.12.80.0/30     B 170     100      >10.12.80.1    100 I

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

```

```
bgp.l3vpn.0: 3 destinations, 3 routes (0 active, 0 holddown, 3 hidden)  
Restart Complete
```

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)  
Restart Complete
```

```
private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route instance

Syntax	show route instance <brief detail summary> <instance-name> <logical-system (all <i>logical-system-name</i>)> <operational>
Syntax (EX Series Switches and QFX Series)	show route instance <brief detail summary> <instance-name> <operational>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.3 for the QFX Series.
Description	Display routing instance information.
Options	<p>none—(Same as brief) Display standard information about all routing instances.</p> <p>brief detail summary—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief. (These options are not available with the operational keyword.)</p> <p>instance-name—(Optional) Display information for all routing instances whose name begins with this string (for example, cust1, cust11, and cust111 are all displayed when you run the show route instance cust1 command).</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p>operational—(Optional) Display operational routing instances.</p>
Required Privilege Level	view
List of Sample Output	show route instance on page 239 show route instance detail (Graceful Restart Complete) on page 239 show route instance detail (Graceful Restart Incomplete) on page 241 show route instance detail (VPLS Routing Instance) on page 242 show route instance operational on page 243 show route instance summary on page 243
Output Fields	Table 20 on page 237 lists the output fields for the show route instance command. Output fields are listed in the approximate order in which they appear.

Table 20: show route instance Output Fields

Field Name	Field Description	Level of Output
Instance or <i>instance-name</i>	Name of the routing instance.	All levels

Table 20: show route instance Output Fields (*continued*)

Field Name	Field Description	Level of Output
Operational Routing Instances	(operational keyword only) Names of all operational routing instances.	—
Type	Type of routing instance: forwarding , l2vpn , no-forwarding , vpls , virtual-router , or vrf .	All levels
State	State of the routing instance: active or inactive .	brief detail none
Interfaces	Name of interfaces belonging to this routing instance.	brief detail none
Restart State	Status of graceful restart for this instance: Pending or Complete .	detail
Path selection timeout	Maximum amount of time, in seconds, remaining until graceful restart is declared complete. The default is 300 .	detail
Tables	Tables (and number of routes) associated with this routing instance.	brief detail none
Route-distinguisher	Unique route distinguisher associated with this routing instance.	detail
Vrf-import	VPN routing and forwarding instance import policy name.	detail
Vrf-export	VPN routing and forwarding instance export policy name.	detail
Vrf-import-target	VPN routing and forwarding instance import target community name.	detail
Vrf-export-target	VPN routing and forwarding instance export target community name.	detail
Fast-reroute-priority	Fast reroute priority setting for a VPLS routing instance: high , medium , or low . The default is low .	detail
Restart State	Restart state: <ul style="list-style-type: none"> • Pending:protocol-name—List of protocols that have not yet completed graceful restart for this routing table. • Complete—All protocols have restarted for this routing table. 	detail
Primary rib	Primary table for this routing instance.	brief none summary
Active/holddown/hidden	Number of active, hold-down, and hidden routes.	All levels

Sample Output

show route instance

```
user@host> show route instance
Instance              Type
Primary RIB
master                forwarding
inet.0                16/0/1
iso.0                 1/0/0
mpls.0                0/0/0
inet6.0               2/0/0
l2circuit.0          0/0/0
__juniper_private1__ forwarding
__juniper_private1__.inet.0 12/0/0
__juniper_private1__.inet6.0 1/0/0
```

show route instance detail (Graceful Restart Complete)

```
user@host> show route instance detail
master:
Router ID: 10.255.14.176
Type: forwarding      State: Active
Restart State: Complete Path selection timeout: 300
Tables:
inet.0                : 17 routes (15 active, 0 holddown, 1 hidden)
Restart Complete
inet.3                : 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
iso.0                 : 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
mpls.0               : 19 routes (19 active, 0 holddown, 0 hidden)
Restart Complete
bgp.l3vpn.0          : 10 routes (10 active, 0 holddown, 0 hidden)
Restart Complete
inet6.0              : 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
bgp.l2vpn.0          : 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
BGP-INET:
Router ID: 10.69.103.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
t3-0/0/0.103
Route-distinguisher: 10.255.14.176:103
Vrf-import: [ BGP-INET-import ]
Vrf-export: [ BGP-INET-export ]
Tables:
BGP-INET.inet.0       : 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
BGP-L:
Router ID: 10.69.104.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
t3-0/0/0.104
Route-distinguisher: 10.255.14.176:104
Vrf-import: [ BGP-L-import ]
Vrf-export: [ BGP-L-export ]
Tables:
BGP-L.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
```

```
BGP-L.mpls.0          : 3 routes (3 active, 0 holddown, 0 hidden)
Restart Complete
L2VPN:
Router ID: 0.0.0.0
Type: l2vpn           State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
  t3-0/0/0.512
Route-distinguisher: 10.255.14.176:512
Vrf-import: [ L2VPN-import ]
Vrf-export: [ L2VPN-export ]
Tables:
  L2VPN.l2vpn.0       : 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
LDP:
Router ID: 10.69.105.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
  t3-0/0/0.105
Route-distinguisher: 10.255.14.176:105
Vrf-import: [ LDP-import ]
Vrf-export: [ LDP-export ]
Tables:
  LDP.inet.0          : 5 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
OSPF:
Router ID: 10.69.101.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
  t3-0/0/0.101
Route-distinguisher: 10.255.14.176:101
Vrf-import: [ OSPF-import ]
Vrf-export: [ OSPF-export ]
Vrf-import-target: [ target:11111
Tables:
  OSPF.inet.0         : 8 routes (7 active, 0 holddown, 0 hidden)
Restart Complete
RIP:
Router ID: 10.69.102.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
  t3-0/0/0.102
Route-distinguisher: 10.255.14.176:102
Vrf-import: [ RIP-import ]
Vrf-export: [ RIP-export ]
Tables:
  RIP.inet.0          : 6 routes (6 active, 0 holddown, 0 hidden)
Restart Complete
STATIC:
Router ID: 10.69.100.1
Type: vrf             State: Active
Restart State: Complete Path selection timeout: 300
Interfaces:
  t3-0/0/0.100
Route-distinguisher: 10.255.14.176:100
Vrf-import: [ STATIC-import ]
Vrf-export: [ STATIC-export ]
Tables:
```



```

STATIC.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

```

**show route instance
detail (Graceful
Restart Incomplete)**

```

user@host> show route instance detail
master:
  Router ID: 10.255.14.176
  Type: forwarding      State: Active
  Restart State: Pending Path selection timeout: 300
  Tables:
    inet.0              : 17 routes (15 active, 1 holddown, 1 hidden)
    Restart Pending: OSPF LDP
    inet.3              : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: OSPF LDP
    iso.0               : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0              : 23 routes (23 active, 0 holddown, 0 hidden)
    Restart Pending: LDP VPN
    bgp.l3vpn.0         : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
    inet6.0             : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0         : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
BGP-INET:
  Router ID: 10.69.103.1
  Type: vrf             State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.255.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0     : 6 routes (5 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
BGP-L:
  Router ID: 10.69.104.1
  Type: vrf             State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.104
  Route-distinguisher: 10.255.14.176:104
  Vrf-import: [ BGP-L-import ]
  Vrf-export: [ BGP-L-export ]
  Tables:
    BGP-L.inet.0        : 6 routes (5 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
    BGP-L.mpls.0        : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn           State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.255.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0       : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: VPN L2VPN

```

```

LDP:
  Router ID: 10.69.105.1
  Type: vrf                      State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.105
  Route-distinguisher: 10.255.14.176:105
  Vrf-import: [ LDP-import ]
  Vrf-export: [ LDP-export ]
  Tables:
    LDP.inet.0                  : 5 routes (4 active, 1 holddown, 0 hidden)
    Restart Pending: OSPF LDP VPN

OSPF:
  Router ID: 10.69.101.1
  Type: vrf                      State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.255.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0                : 8 routes (7 active, 1 holddown, 0 hidden)
    Restart Pending: OSPF VPN

RIP:
  Router ID: 10.69.102.1
  Type: vrf                      State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.255.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0                 : 8 routes (6 active, 2 holddown, 0 hidden)
    Restart Pending: RIP VPN

STATIC:
  Router ID: 10.69.100.1
  Type: vrf                      State: Active
  Restart State: Pending Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.255.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0             : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Pending: VPN

```

show route instance detail (VPLS Routing Instance)

```

user@host> show route instance detail test-vpls
test-vpls:
  Router ID: 0.0.0.0
  Type: vpls                      State: Active
  Interfaces:
    lsi.1048833
    lsi.1048832
    fe-0/1/0.513
  Route-distinguisher: 10.255.37.65:1
  Vrf-import: [ __vrf-import-test-vpls-internal__ ]
  Vrf-export: [ __vrf-export-test-vpls-internal__ ]
  Vrf-import-target: [ target:300:1 ]

```

```

Vrf-export-target: [ target:300:1 ]
Fast-reroute-priority: high
Tables:
    test-vpls.l2vpn.0          : 3 routes (3 active, 0 holddown, 0 hidden)

```

show route instance operational

```

user@host> show route instance operational
Operational Routing Instances:

```

```

master
default

```

show route instance summary

```

user@host> show route instance summary

```

Instance	Type	Primary rib	Active/holddown/hidden
master	forwarding	inet.0	15/0/1
		iso.0	1/0/0
		mpls.0	35/0/0
		l3vpn.0	0/0/0
		inet6.0	2/0/0
		l2vpn.0	0/0/0
		l2circuit.0	0/0/0
BGP-INET	vrf	BGP-INET.inet.0	5/0/0
		BGP-INET.iso.0	0/0/0
		BGP-INET.inet6.0	0/0/0
BGP-L	vrf	BGP-L.inet.0	5/0/0
		BGP-L.iso.0	0/0/0
		BGP-L.mpls.0	4/0/0
		BGP-L.inet6.0	0/0/0
L2VPN	l2vpn	L2VPN.inet.0	0/0/0
		L2VPN.iso.0	0/0/0
		L2VPN.inet6.0	0/0/0
		L2VPN.l2vpn.0	2/0/0
LDP	vrf	LDP.inet.0	4/0/0
		LDP.iso.0	0/0/0
		LDP.mpls.0	0/0/0
		LDP.inet6.0	0/0/0
		LDP.l2circuit.0	0/0/0
OSPF	vrf	OSPF.inet.0	7/0/0
		OSPF.iso.0	0/0/0
		OSPF.inet6.0	0/0/0
RIP	vrf	RIP.inet.0	6/0/0
		RIP.iso.0	0/0/0
		RIP.inet6.0	0/0/0
STATIC	vrf	STATIC.inet.0	4/0/0
		STATIC.iso.0	0/0/0
		STATIC.inet6.0	0/0/0

show route next-hop

Syntax	<code>show route next-hop <i>next-hop</i></code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Syntax (EX Series Switches)	<code>show route next-hop <i>next-hop</i></code> <code><brief detail extensive terse></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display the entries in the routing table that are being sent to the specified next-hop address.
Options	brief detail extensive terse —(Optional) Display the specified level of output. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system. <i>next-hop</i> —Next-hop address.
Required Privilege Level	view
List of Sample Output	show route next-hop on page 245 show route next-hop detail on page 245 show route next-hop extensive on page 247 show route next-hop terse on page 249
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route next-hop

```

user@host> show route next-hop 192.168.71.254

inet.0: 18 destinations, 18 routes (17 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.10.0.0/16      *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
10.209.0.0/16    *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
172.16.0.0/12    *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
192.168.0.0/16   *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
192.168.102.0/23 *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
207.17.136.0/24  *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0
207.17.136.192/32 *[Static/5] 06:26:25
                  > to 192.168.71.254 via fxp0.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

red.inet.0: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```

show route next-hop detail

```

user@host> show route next-hop 192.168.71.254 detail

inet.0: 18 destinations, 18 routes (17 active, 0 holddown, 1 hidden)
Restart Complete
10.10.0.0/16 (1 entry, 1 announced)
    *Static Preference: 5
        Next-hop reference count: 36
        Next hop: 192.168.71.254 via fxp0.0, selected
        State: <Active NoReadvrt Int Ext>
        Local AS: 1
        Age: 6:27:41
        Task: RT
        Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
        AS path: I
10.209.0.0/16 (1 entry, 1 announced)
    *Static Preference: 5
        Next-hop reference count: 36
        Next hop: 192.168.71.254 via fxp0.0, selected
        State: <Active NoReadvrt Int Ext>

```

```
Local AS:      1
Age: 6:27:41
Task: RT
Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
AS path: I

172.16.0.0/12 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 36
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS:      1
    Age: 6:27:41
    Task: RT
    Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
    AS path: I

192.168.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 36
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS:      1
    Age: 6:27:41
    Task: RT
    Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
    AS path: I

192.168.102.0/23 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 36
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS:      1
    Age: 6:27:41
    Task: RT
    Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
    AS path: I

207.17.136.0/24 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 36
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS:      1
    Age: 6:27:41
    Task: RT
    Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
    AS path: I

207.17.136.192/32 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 36
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS:      1
    Age: 6:27:41
    Task: RT
    Announcement bits (3): 0-KRT 3-Resolve tree 1 5-Resolve tree 2
    AS path: I
```

```

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

red.inet.0: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```

show route next-hop extensive

```

user@host> show route next-hop 192.168.71.254 extensive

inet.0: 18 destinations, 18 routes (17 active, 0 holddown, 1 hidden)
10.10.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.10.0.0/16 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

10.209.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.209.0.0/16 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

172.16.0.0/12 (1 entry, 1 announced)
TSI:
KRT in-kernel 172.16.0.0/12 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

192.168.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 192.168.0.0/16 -> {192.168.71.254}

```

```
*Static Preference: 5
  Next-hop reference count: 22
  Next hop: 192.168.71.254 via fxp0.0, selected
  State: <Active NoReadvrt Int Ext>
  Local AS: 69
  Age: 2:02:28
  Task: RT
  Announcement bits (1): 0-KRT
  AS path: I

192.168.102.0/23 (1 entry, 1 announced)
TSI:
KRT in-kernel 192.168.102.0/23 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

207.17.136.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 207.17.136.0/24 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

207.17.136.192/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 207.17.136.192/32 -> {192.168.71.254}
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 2:02:28
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

green.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```



```
red.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route next-hop terse

```
user@host> show route next-hop 192.168.71.254 terse
```

```
inet.0: 25 destinations, 26 routes (24 active, 0 holddown, 1 hidden)
```

```
Restart Complete
```

```
+ = Active Route, - = Last Active, * = Both
```

A	Destination	P	Prf	Metric 1	Metric 2	Next hop	AS path
*	10.10.0.0/16	S	5			>192.168.71.254	
*	10.209.0.0/16	S	5			>192.168.71.254	
*	172.16.0.0/12	S	5			>192.168.71.254	
*	192.168.0.0/16	S	5			>192.168.71.254	
*	192.168.102.0/23	S	5			>192.168.71.254	
*	207.17.136.0/24	S	5			>192.168.71.254	
*	207.17.136.192/32	S	5			>192.168.71.254	

```
private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
```

```
red.inet.0: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
Restart Complete
```

```
private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route output

Syntax	<code>show route output (address <i>ip-address</i> interface <i>interface-name</i>)</code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Syntax (EX Series Switches)	<code>show route output (address <i>ip-address</i> interface <i>interface-name</i>)</code> <code><brief detail extensive terse></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Display the entries in the routing table learned through static routes and interior gateway protocols that are to be sent out the interface with either the specified IP address or specified name.</p> <p>To view routes advertised to a neighbor or received from a neighbor for the BGP protocol, use the show route advertising-protocol bgp and show route receive-protocol bgp commands instead.</p>
Options	<p>address <i>ip-address</i>—Display entries in the routing table that are to be sent out the interface with the specified IP address.</p> <p>brief detail extensive terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.</p> <p>interface <i>interface-name</i>—Display entries in the routing table that are to be sent out the interface with the specified name.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route output address on page 251 show route output address detail on page 251 show route output address extensive on page 251 show route output address terse on page 251 show route output interface on page 252 show route output interface detail on page 252 show route output interface extensive on page 253 show route output interface terse on page 253
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

show route output address

```
user@host> show route output address 36.1.1.1/24

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

36.1.1.0/24          *[Direct/0] 00:19:56
                    > via so-0/1/2.0
                    [OSPF/10] 00:19:55, metric 1
                    > via so-0/1/2.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route output address detail

```
user@host> show route output address 36.1.1.1 detail

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
36.1.1.0/24 (2 entries, 0 announced)
    *Direct Preference: 0
        Next hop type: Interface
        Next-hop reference count: 1
        Next hop: via so-0/1/2.0, selected
        State: <Active Int>
        Age: 23:00
        Task: IF
        AS path: I
    OSPF Preference: 10
        Next-hop reference count: 1
        Next hop: via so-0/1/2.0, selected
        State: <Int>
        Inactive reason: Route Preference
        Age: 22:59      Metric: 1
        Area: 0.0.0.0
        Task: OSPF
        AS path: I

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route output address extensive

The output for the **show route output address extensive** command is identical to that of the **show route output address detail** command. For sample output, see [show route output address detail on page 251](#).

**show route output
address terse**

```

user@host> show route output address 36.1.1.1 terse

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf  Metric 1   Metric 2   Next hop      AS path
* 36.1.1.0/24      D   0           1           >so-0/1/2.0
                   0  10           1           >so-0/1/2.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```

**show route output
interface**

```

user@host> show route output interface so-0/1/2.0

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.71.240/32  *[OSPF/10] 00:13:00, metric 2
                   via so-0/1/2.0
                   > via so-0/3/2.0
10.255.71.241/32  *[OSPF/10] 00:13:10, metric 1
                   > via so-0/1/2.0
14.1.1.0/24       *[OSPF/10] 00:05:11, metric 3
                   to 35.1.1.2 via ge-3/1/0.0
                   > via so-0/1/2.0
                   via so-0/3/2.0
16.1.1.0/24       *[OSPF/10] 00:13:10, metric 2
                   > via so-0/1/2.0
36.1.1.0/24       *[Direct/0] 00:13:21
                   > via so-0/1/2.0
                   [OSPF/10] 00:13:20, metric 1
                   > via so-0/1/2.0

private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```

**show route output
interface detail**

```

user@host> show route output interface so-0/1/2.0 detail

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
10.255.71.240/32 (1 entry, 1 announced)
    *OSPF   Preference: 10
            Next-hop reference count: 2
            Next hop: via so-0/1/2.0
            Next hop: via so-0/3/2.0, selected
            State: <Active Int>
            Age: 14:52      Metric: 2

```

```

Area: 0.0.0.0
Task: OSPF
Announcement bits (1): 0-KRT
AS path: I

10.255.71.241/32 (1 entry, 1 announced)
*OSPF Preference: 10
Next-hop reference count: 4
Next hop: via so-0/1/2.0, selected
State: <Active Int>
Age: 15:02 Metric: 1
Area: 0.0.0.0
Task: OSPF
Announcement bits (1): 0-KRT
AS path: I

...

```

show route output interface extensive

The output for the **show route output interface extensive** command is identical to that of the **show route output interface detail** command. For sample output, see [show route output interface detail on page 252](#).

show route output interface terse

```
user@host> show route output interface so-0/1/2.0 terse
```

```
inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

A	Destination	P	Prf	Metric 1	Metric 2	Next hop	AS path
*	10.255.71.240/32	0	10	2		so-0/1/2.0	
						>so-0/3/2.0	
*	10.255.71.241/32	0	10	1		>so-0/1/2.0	
*	14.1.1.0/24	0	10	3		35.1.1.2	
						>so-0/1/2.0	
						so-0/3/2.0	
*	16.1.1.0/24	0	10	2		>so-0/1/2.0	
*	36.1.1.0/24	D	0			>so-0/1/2.0	
		0	10	1		>so-0/1/2.0	

```
private1___.inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
```

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
private1___.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

show route protocol

Syntax	<code>show route protocol <i>protocol</i></code> <code><brief detail extensive terse></code> <code><logical-system (all <i>logical-system-name</i>)></code>
Syntax (EX Series Switches)	<code>show route protocol <i>protocol</i></code> <code><brief detail extensive terse></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. ospf2 and ospf3 options introduced in Junos OS Release 9.2. ospf2 and ospf3 options introduced in Junos OS Release 9.2 for EX Series switches. flow option introduced in Junos OS Release 10.0. flow option introduced in Junos OS Release 10.0 for EX Series switches.
Description	Display the route entries in the routing table that were learned from a particular protocol.
Options	brief detail extensive terse —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief . logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system. <i>protocol</i> —Protocol from which the route was learned: <ul style="list-style-type: none">• access—Access route for use by DHCP application• access-internal—Access-internal route for use by DHCP application• aggregate—Locally generated aggregate route• arp—Route learned through the Address Resolution Protocol• atmvpn—Asynchronous Transfer Mode virtual private network• bgp—Border Gateway Protocol• ccc—Circuit cross-connect• direct—Directly connected route• dvmrp—Distance Vector Multicast Routing Protocol• esis—End System-to-Intermediate System• flow—Locally defined flow-specification route• frr—Precomputed protection route or backup route used when a link goes down• isis—Intermediate System-to-Intermediate System• ldp—Label Distribution Protocol• l2circuit—Layer 2 circuit• l2vpn—Layer 2 virtual private network

- **local**—Local address
- **mpls**—Multiprotocol Label Switching
- **msdp**—Multicast Source Discovery Protocol
- **ospf**—Open Shortest Path First versions 2 and 3
- **ospf2**—Open Shortest Path First versions 2 only
- **ospf3**—Open Shortest Path First version 3 only
- **pim**—Protocol Independent Multicast
- **rip**—Routing Information Protocol
- **ripng**—Routing Information Protocol next generation
- **rsvp**—Resource Reservation Protocol
- **rtarget**—Local route target virtual private network
- **static**—Statically defined route
- **tunnel**—Dynamic tunnel
- **vpn**—Virtual private network



NOTE: EX Series switches run a subset of these protocols. See the switch CLI for details.

Required Privilege Level	view
List of Sample Output	show route protocol access on page 257 show route protocol access-internal extensive on page 257 show route protocol arp on page 257 show route protocol bgp on page 258 show route protocol bgp detail on page 258 show route protocol bgp extensive on page 258 show route protocol bgp terse on page 259 show route protocol direct on page 259 show route protocol frr on page 259 show route protocol l2circuit detail on page 260 show route protocol l2vpn extensive on page 261 show route protocol ldp on page 261 show route protocol ldp extensive on page 262 show route protocol ospf (Layer 3 VPN) on page 263 show route protocol ospf detail on page 263 show route protocol rip on page 264 show route protocol rip detail on page 264 show route protocol ripng table inet6 on page 264 show route protocol static detail on page 264

Output Fields For information about output fields, see the output field tables for the [show route](#) command, the [show route detail](#) command, the [show route extensive](#) command, or the [show route terse](#) command.

Sample Output

show route protocol access

```
user@host> show route protocol access
inet.0: 30380 destinations, 30382 routes (30379 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

13.160.0.3/32      *[Access/13] 00:00:09
                  > to 13.160.0.2 via fe-0/0/0.0
13.160.0.4/32      *[Access/13] 00:00:09
                  > to 13.160.0.2 via fe-0/0/0.0
13.160.0.5/32      *[Access/13] 00:00:09
                  > to 13.160.0.2 via fe-0/0/0.0
```

show route protocol access-internal extensive

```
user@host> show route protocol access-internal 13.160.0.19 extensive
inet.0: 100020 destinations, 100022 routes (100019 active, 0 holddown, 1 hidden)
13.160.0.19/32 (1 entry, 1 announced)
TSI:
KRT in-kerne1 13.160.0.19/32 -> {13.160.0.2}
    *Access-internal Preference: 12
        Next-hop reference count: 200000
        Next hop: 13.160.0.2 via fe-0/0/0.0, selected
        State: <Active Int>
    Age: 36
        Task: RPD Unix Domain Server./var/run/rpd_serv.local
        Announcement bits (1): 0-KRT
        AS path: I
```

show route protocol arp

```
user@host> show route protocol arp
inet.0: 43 destinations, 43 routes (42 active, 0 holddown, 1 hidden)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

cust1.inet.0: 1033 destinations, 2043 routes (1033 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

20.20.1.3/32      [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.4/32      [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.5/32      [ARP/4294967293] 00:04:32, from 20.20.1.1
                  Unusable
20.20.1.6/32      [ARP/4294967293] 00:04:34, from 20.20.1.1
                  Unusable
20.20.1.7/32      [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.8/32      [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.9/32      [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.10/32     [ARP/4294967293] 00:04:35, from 20.20.1.1
                  Unusable
20.20.1.11/32     [ARP/4294967293] 00:04:33, from 20.20.1.1
                  Unusable
20.20.1.12/32     [ARP/4294967293] 00:04:33, from 20.20.1.1
                  Unusable
20.20.1.13/32     [ARP/4294967293] 00:04:33, from 20.20.1.1
                  Unusable
...
```

**show route protocol
bgp**

```

user@host> show route protocol bgp 192.168.64.0/21
inet.0: 335832 destinations, 335833 routes (335383 active, 0 holddown, 450 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.64.0/21      *[BGP/170] 6d 10:41:16, localpref 100, from 192.168.69.71
                    AS path: 10458 14203 2914 4788 4788 I
                    > to 192.168.167.254 via fxp0.0

```

**show route protocol
bgp detail**

```

user@host> show route protocol bgp 66.117.63.0/24 detail
inet.0: 335805 destinations, 335806 routes (335356 active, 0 holddown, 450 hidden)
66.117.63.0/24 (1 entry, 1 announced)
    *BGP      Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 1006436
                Source: 192.168.69.71
                Next hop type: Router, Next hop index: 324
                Next hop: 192.168.167.254 via fxp0.0, selected
                Protocol next hop: 192.168.69.71
                Indirect next hop: 8e166c0 342
                State: <Active Ext>
                Local AS: 69 Peer AS: 10458
                Age: 6d 10:42:42 Metric2: 0
                Task: BGP_10458.192.168.69.71+179
                Announcement bits (3): 0-KRT 2-BGP RT Background 3-Resolve tree

1

    AS path: 10458 14203 2914 4788 4788 I
    Communities: 2914:410 2914:2403 2914:3400
    Accepted
    Localpref: 100
    Router ID: 207.17.136.192

```

**show route protocol
bgp extensive**

```

user@host> show route protocol bgp 192.168.64.0/21 extensive

inet.0: 335827 destinations, 335828 routes (335378 active, 0 holddown, 450 hidden)
192.168.64.0/21 (1 entry, 1 announced)
TSI:
KRT in-kernel 1.9.0.0/16 -> {indirect(342)}
Page 0 idx 1 Type 1 val db31a80
    Nexthop: Self
    AS path: [69] 10458 14203 2914 4788 4788 I
    Communities: 2914:410 2914:2403 2914:3400
Path 1.9.0.0 from 192.168.69.71 Vector len 4. Val: 1
    *BGP      Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 1006502
                Source: 192.168.69.71
                Next hop type: Router, Next hop index: 324
                Next hop: 192.168.167.254 via fxp0.0, selected
                Protocol next hop: 192.168.69.71
                Indirect next hop: 8e166c0 342
                State: <Active Ext>
                Local AS: 69 Peer AS: 10458
                Age: 6d 10:44:45 Metric2: 0
                Task: BGP_10458.192.168.69.71+179
                Announcement bits (3): 0-KRT 2-BGP RT Background 3-Resolve tree

1

    AS path: 10458 14203 2914 4788 4788 I
    Communities: 2914:410 2914:2403 2914:3400

```

```

Accepted
Localpref: 100
Router ID: 207.17.136.192
Indirect next hops: 1
  Protocol next hop: 192.168.69.71
  Indirect next hop: 8e166c0 342
  Indirect path forwarding next hops: 1
    Next hop type: Router
    Next hop: 192.168.167.254 via fxp0.0
  192.168.0.0/16 Originating RIB: inet.0
  Node path count: 1
  Forwarding nexthops: 1
    Nexthop: 192.168.167.254 via fxp0.0

```

show route protocol bgp terse

```
user@host> show route protocol bgp 192.168.64.0/21 terse
```

```

inet.0: 24 destinations, 32 routes (23 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf   Metric 1   Metric 2   Next hop      AS path
192.168.64.0/21   B 170      100           >100.1.3.2    10023 21 I

```

show route protocol direct

```
user@host> show route protocol direct
```

```

inet.0: 335843 destinations, 335844 routes (335394 active, 0 holddown, 450 hidden)
+ = Active Route, - = Last Active, * = Both

8.8.8.0/24          *[Direct/0] 17w0d 10:31:49
> via fe-1/3/1.0
10.255.165.1/32     *[Direct/0] 25w4d 04:13:18
> via lo0.0
30.30.30.0/24       *[Direct/0] 17w0d 23:06:26
> via fe-1/3/2.0
192.168.164.0/22    *[Direct/0] 25w4d 04:13:20
> via fxp0.0

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.0102.5516.5001/152
*[Direct/0] 25w4d 04:13:21
> via lo0.0

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

abcd::10:255:165:1/128
*[Direct/0] 25w4d 04:13:21
> via lo0.0
fe80::2a0:a5ff:fe12:ad7/128
*[Direct/0] 25w4d 04:13:21
> via lo0.0

```

show route protocol frr

```
user@host> show route protocol frr
```

```

inet.0: 43 destinations, 43 routes (42 active, 0 holddown, 1 hidden)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

cust1.inet.0: 1033 destinations, 2043 routes (1033 active, 0 holddown, 0 hidden)

```

+ = Active Route, - = Last Active, * = Both

```

20.20.1.3/32      *[FRR/200] 00:05:38, from 20.20.1.1
                  > to 20.20.1.3 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.4/32      *[FRR/200] 00:05:38, from 20.20.1.1
                  > to 20.20.1.4 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.5/32      *[FRR/200] 00:05:35, from 20.20.1.1
                  > to 20.20.1.5 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.6/32      *[FRR/200] 00:05:37, from 20.20.1.1
                  > to 20.20.1.6 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.7/32      *[FRR/200] 00:05:38, from 20.20.1.1
                  > to 20.20.1.7 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.8/32      *[FRR/200] 00:05:38, from 20.20.1.1
                  > to 20.20.1.8 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.9/32      *[FRR/200] 00:05:38, from 20.20.1.1
                  > to 20.20.1.9 via ge-4/1/0.0
                  to 10.10.15.1 via ge-0/2/4.0, Push 16, Push 299792(top)
20.20.1.10/32     *[FRR/200] 00:05:38, from 20.20.1.1
...

```

show route protocol l2circuit detail

user@host> show route protocol l2circuit detail

```

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
100000 (1 entry, 1 announced)
  *L2CKT Preference: 7
    Next hop: via ge-2/0/0.0, selected
    Label operation: Pop      Offset: 4
    State: <Active Int>
    Local AS: 99
    Age: 9:52
    Task: Common L2 VC
    Announcement bits (1): 0-KRT
    AS path: I

ge-2/0/0.0 (1 entry, 1 announced)
  *L2CKT Preference: 7
    Next hop: via so-1/1/2.0 weight 1, selected
    Label-switched-path my-lsp
    Label operation: Push 100000, Push 100000(top)[0] Offset: -4
    Protocol next hop: 10.245.255.63
    Push 100000 Offset: -4
    Indirect next hop: 86af0c0 298
    State: <Active Int>
    Local AS: 99
    Age: 9:52
    Task: Common L2 VC
    Announcement bits (2): 0-KRT 1-Common L2 VC
    AS path: I

l2circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

10.245.255.63:CtrlWord:4:3:Local/96 (1 entry, 1 announced)
  *L2CKT Preference: 7
    Next hop: via so-1/1/2.0 weight 1, selected
    Label-switched-path my-lsp

```

```

Label operation: Push 100000[0]
Protocol next hop: 10.245.255.63 Indirect next hop: 86af000 296
State: <Active Int>
Local AS: 99
Age: 10:21
Task: 12 circuit
Announcement bits (1): 0-LDP
AS path: I
VC Label 100000, MTU 1500, VLAN ID 512

```

show route protocol l2vpn extensive

```

user@host> show route protocol l2vpn extensive

inet.0: 14 destinations, 15 routes (13 active, 0 holddown, 1 hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
800001 (1 entry, 1 announced)
TSI:
KRT in-kernel 800001 /36 -> {so-0/0/0.0}
    *L2VPN Preference: 7
      Next hop: via so-0/0/0.0 weight 49087 balance 97%, selected
      Label operation: Pop Offset: 4
      State: <Active Int>
      Local AS: 69
      Age: 7:48
      Task: Common L2 VC
      Announcement bits (1): 0-KRT
      AS path: I

so-0/0/0.0 (1 entry, 1 announced)
TSI:
KRT in-kernel so-0/0/0.0 /16 -> {indirect(288)}
    *L2VPN Preference: 7
      Next hop: via so-0/0/1.0, selected
      Label operation: Push 800000 Offset: -4
      Protocol next hop: 10.255.14.220
      Push 800000 Offset: -4
      Indirect next hop: 85142a0 288
      State: <Active Int>
      Local AS: 69
      Age: 7:48
      Task: Common L2 VC
      Announcement bits (2): 0-KRT 1-Common L2 VC
      AS path: I
      Communities: target:69:1 Layer2-info: encaps:PPP,
      control flags:2, mtu: 0

```

show route protocol ldp

```

user@host> show route protocol ldp

inet.0: 12 destinations, 13 routes (12 active, 0 holddown, 0 hidden)

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.16.1/32    *[LDP/9] 1d 23:03:35, metric 1
                  > via t1-4/0/0.0, Push 100000
192.168.17.1/32    *[LDP/9] 1d 23:03:35, metric 1
                  > via t1-4/0/0.0

```

```
private1___.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
100064          *[LDP/9] 1d 23:03:35, metric 1
                 > via t1-4/0/0.0, Pop
100064(S=0)     *[LDP/9] 1d 23:03:35, metric 1
                 > via t1-4/0/0.0, Pop
100080          *[LDP/9] 1d 23:03:35, metric 1
                 > via t1-4/0/0.0, Swap 100000
```

show route protocol ldp extensive

```
user@host> show route protocol ldp extensive
```

```
192.168.16.1/32 (1 entry, 1 announced)
```

```
State: <FlashAll>
```

```
*LDP Preference: 9
```

```
Next-hop reference count: 3
```

```
Next hop: via t1-4/0/0.0, selected
```

```
Label operation: Push 100000
```

```
State: <Active Int>
```

```
Local AS: 65500
```

```
Age: 1d 23:03:58 Metric: 1
```

```
Task: LDP
```

```
Announcement bits (2): 0-Resolve tree 1 2-Resolve tree 2
```

```
AS path: I
```

```
192.168.17.1/32 (1 entry, 1 announced)
```

```
State: <FlashAll>
```

```
*LDP Preference: 9
```

```
Next-hop reference count: 3
```

```
Next hop: via t1-4/0/0.0, selected
```

```
State: <Active Int>
```

```
Local AS: 65500
```

```
Age: 1d 23:03:58 Metric: 1
```

```
Task: LDP
```

```
Announcement bits (2): 0-Resolve tree 1 2-Resolve tree 2
```

```
AS path: I
```

```
private1___.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

```
100064 (1 entry, 1 announced)
```

```
TSI:
```

```
KRT in-kerne1 100064 /36 -> {t1-4/0/0.0}
```

```
*LDP Preference: 9
```

```
Next-hop reference count: 2
```

```
Next hop: via t1-4/0/0.0, selected
```

```
State: <Active Int>
```

```
Local AS: 65500
```

```
Age: 1d 23:03:58 Metric: 1
```

```
Task: LDP
```

```
Announcement bits (1): 0-KRT
```

```
AS path: I
```

```
Prefixes bound to route: 192.168.17.1/32
```

```
100064(S=0) (1 entry, 1 announced)
```

```
TSI:
```

```
KRT in-kerne1 100064 /40 -> {t1-4/0/0.0}
```

```
*LDP Preference: 9
```

```

Next-hop reference count: 2
Next hop: via t1-4/0/0.0, selected
Label operation: Pop
State: <Active Int>
Local AS: 65500
Age: 1d 23:03:58      Metric: 1
Task: LDP
Announcement bits (1): 0-KRT
AS path: I

100080 (1 entry, 1 announced)
TSI:
KRT in-kernel 100080 /36 -> {t1-4/0/0.0}
    *LDP      Preference: 9
              Next-hop reference count: 2
              Next hop: via t1-4/0/0.0, selected
              Label operation: Swap 100000
              State: <Active Int>
              Local AS: 65500
              Age: 1d 23:03:58      Metric: 1
              Task: LDP
              Announcement bits (1): 0-KRT
              AS path: I
              Prefixes bound to route: 192.168.16.1/32

```

show route protocol ospf (Layer 3 VPN)

```

user@host> show route protocol ospf
inet.0: 40 destinations, 40 routes (39 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.39.1.4/30      *[OSPF/10] 00:05:18, metric 4
                  > via t3-3/2/0.0
10.39.1.8/30      [OSPF/10] 00:05:18, metric 2
                  > via t3-3/2/0.0
10.255.14.171/32  *[OSPF/10] 00:05:18, metric 4
                  > via t3-3/2/0.0
10.255.14.179/32  *[OSPF/10] 00:05:18, metric 2
                  > via t3-3/2/0.0
224.0.0.5/32      *[OSPF/10] 20:25:55, metric 1

VPN-AB.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.39.1.16/30     [OSPF/10] 00:05:43, metric 1
                  > via so-0/2/2.0
10.255.14.173/32  *[OSPF/10] 00:05:43, metric 1
                  > via so-0/2/2.0
224.0.0.5/32      *[OSPF/10] 20:26:20, metric 1

```

show route protocol ospf detail

```

user@host> show route protocol ospf detail
VPN-AB.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.39.1.16/30 (2 entries, 0 announced)
    OSPF      Preference: 10
              Nexthop: via so-0/2/2.0, selected
              State: <Int>
              Inactive reason: Route Preference
              Age: 6:25      Metric: 1
              Area: 0.0.0.0
              Task: VPN-AB-OSPF

```

```
AS path: I
Communities: Route-Type:0.0.0.0:1:0
```

```
...
```

show route protocol rip

```
user@host> show route protocol rip
inet.0: 26 destinations, 27 routes (25 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

VPN-AB.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.14.177/32    *[RIP/100] 20:24:34, metric 2
                  > to 10.39.1.22 via t3-0/2/2.0
224.0.0.9/32      *[RIP/100] 00:03:59, metric 1
```

show route protocol rip detail

```
user@host> show route protocol rip detail
inet.0: 26 destinations, 27 routes (25 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

VPN-AB.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.14.177/32 (1 entry, 1 announced)
    *RIP      Preference: 100
              Nexthop: 10.39.1.22 via t3-0/2/2.0, selected
              State: <Active Int>
              Age: 20:25:02   Metric: 2
              Task: VPN-AB-RIPv2
              Announcement bits (2): 0-KRT 2-BGP.0.0.0.0+179
              AS path: I
              Route learned from 10.39.1.22 expires in 96 seconds
```

show route protocol ripng table inet6

```
user@host> show route protocol ripng table inet6
inet6.0: 4215 destinations, 4215 routes (4214 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

1111::1/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
1111::2/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
1111::3/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
1111::4/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
1111::5/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
1111::6/128      *[RIPng/100] 02:13:33, metric 2
                  > to fe80::2a0:a5ff:fe3d:56 via t3-0/2/0.0
```

show route protocol static detail

```
user@host> show route protocol static detail
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
10.5.0.0/16 (1 entry, 1 announced)
    *Static Preference: 5
      Next hop type: Router, Next hop index: 324
      Address: 0x9274010
      Next-hop reference count: 27
      Next hop: 192.168.187.126 via fxp0.0, selected
      Session Id: 0x0
      State: <Active NoReadvrt Int Ext>
```



```
Age: 7w3d 21:24:25
Validation State: unverified
Task: RT
Announcement bits (1): 0-KRT
AS path: I

10.10.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next hop type: Router, Next hop index: 324
    Address: 0x9274010
    Next-hop reference count: 27
    Next hop: 192.168.187.126 via fxp0.0, selected
    Session Id: 0x0
    State: <Active NoReadvrt Int Ext>
    Age: 7w3d 21:24:25
    Validation State: unverified
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

10.13.10.0/23 (1 entry, 1 announced)
  *Static Preference: 5
    Next hop type: Router, Next hop index: 324
    Address: 0x9274010
    Next-hop reference count: 27
    Next hop: 192.168.187.126 via fxp0.0, selected
    Session Id: 0x0
    State: <Active NoReadvrt Int Ext>
    Age: 7w3d 21:24:25
    Validation State: unverified
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I
```

show route receive-protocol

Syntax	show route receive-protocol <i>protocol neighbor-address</i> <brief detail extensive terse> <logical-system (all <i>logical-system-name</i>)>	
Syntax (EX Series Switches)	show route receive-protocol <i>protocol neighbor-address</i> <brief detail extensive terse>	
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.	
Description	Display the routing information as it was received through a particular neighbor using a particular dynamic routing protocol.	
Options	brief detail extensive terse —(Optional) Display the specified level of output. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system. <i>protocol neighbor-address</i> —Protocol transmitting the route (bgp , dvmrp , msdp , pim , rip , or ripng) and address of the neighboring router from which the route entry was received.	
Additional Information	The output displays the selected routes and the attributes with which they were received, but does not show the effects of import policy on the routing attributes.	
Required Privilege Level	view	
List of Sample Output	show route receive-protocol bgp on page 269 show route receive-protocol bgp extensive on page 269 show route receive-protocol bgp table extensive on page 269 show route receive-protocol bgp logical-system extensive on page 269 show route receive-protocol bgp detail (Layer 2 VPN) on page 270 show route receive-protocol bgp extensive (Layer 2 VPN) on page 271 show route receive-protocol bgp (Layer 3 VPN) on page 272 show route receive-protocol bgp detail (Layer 3 VPN) on page 272 show route receive-protocol bgp extensive (Layer 3 VPN) on page 273	
Output Fields	Table 21 on page 266 describes the output fields for the show route receive-protocol command. Output fields are listed in the approximate order in which they appear.	

Table 21: show route receive-protocol Output Fields

Field Name	Field Description	Level of Output
<i>routing-table-name</i>	Name of the routing table—for example, inet.0.	All levels
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.	All levels

Table 21: show route receive-protocol Output Fields (*continued*)

Field Name	Field Description	Level of Output
number routes	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> • active • holddown (routes that are in pending state before being declared inactive) • hidden (routes that are not used because of a routing policy) 	All levels
Prefix	Destination prefix.	none brief
MED	Multiple exit discriminator value included in the route.	none brief
destination-prefix (entry, announced)	Destination prefix. The entry value is the number of routes for this destination, and the announced value is the number of routes being announced for this destination.	detail extensive
Route Distinguisher	64-bit prefix added to IP subnets to make them unique.	detail extensive
Label-Base, range	First label in a block of labels and label block size. A remote PE routing device uses this first label when sending traffic toward the advertising PE routing device.	detail extensive
VPN Label	Virtual private network (VPN) label. Packets are sent between CE and PE routing devices by advertising VPN labels. VPN labels transit over either an RSVP or an LDP label-switched path (LSP) tunnel.	detail extensive
Next hop	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.	All levels
Localpref or Lclpref	Local preference value included in the route.	All levels

Table 21: show route receive-protocol Output Fields (*continued*)

Field Name	Field Description	Level of Output
AS path	<p>Autonomous system (AS) path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the number that precedes the AS path. This number represents the number of ASs present in the AS path, when calculated as defined in RFC 4271. This value is used the AS-path merge process, as defined in RFC 4893. • []—If more than one AS number is configured on the router, or if AS path prepending is configured, brackets enclose the local AS number associated with the AS path. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p>	All levels
Cluster list	(For route reflected output only) Cluster ID sent by the route reflector.	detail extensive
Originator ID	(For route reflected output only) Address of routing device that originally sent the route to the route reflector.	detail extensive
Communities	Community path attribute for the route. See the Output Field table in the show route detail command for all possible values for this field.	detail extensive
AIGP	Accumulated interior gateway protocol (AIGP) BGP attribute.	detail extensive
Attrset AS	Number, local preference, and path of the AS that originated the route. These values are stored in the Attrset attribute at the originating routing device.	detail extensive
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).	detail extensive
control flags	Control flags: none or Site Down .	detail extensive
mtu	Maximum transmission unit (MTU) of the Layer 2 circuit.	detail extensive

Sample Output

**show route
receive-protocol bgp**

```
user@host> show route receive-protocol bgp 10.255.245.215

inet.0: 28 destinations, 33 routes (27 active, 0 holddown, 1 hidden)
Prefix          Next hop          MED      Lclpref    AS path
10.22.1.0/24     10.255.245.215    0        100        I
10.22.2.0/24     10.255.245.215    0        100        I
```

**show route
receive-protocol bgp
extensive**

```
user@host> show route receive-protocol bgp 10.255.245.63 extensive

inet.0: 244 destinations, 244 routes (243 active, 0 holddown, 1 hidden)
Prefix          Next hop          MED      Lclpref    AS path
1.1.1.0/24 (1 entry, 1 announced)
  Next hop: 10.0.50.3
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.45
165.3.0.0/16 (1 entry, 1 announced)
  Next hop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.68
165.4.0.0/16 (1 entry, 1 announced)
  Next hop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.45
195.1.2.0/24 (1 entry, 1 announced)
  Next hop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.68
inet.2: 63 destinations, 63 routes (63 active, 0 holddown, 0 hidden)
Prefix          Next hop          MED      Lclpref    AS path
inet.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Prefix          Next hop          MED      Lclpref    AS path
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix          Next hop          MED      Lclpref    AS path
mpls.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
```

**show route
receive-protocol bgp
table extensive**

```
user@host> show route receive-protocol bgp 207.17.136.192 table inet.0 66.117.68.0/24 extensive

inet.0: 227315 destinations, 227316 routes (227302 active, 0 holddown, 13 hidden)
* 66.117.63.0/24 (1 entry, 1 announced)
  Nexthop: 207.17.136.29
  Localpref: 100
  AS path: AS2 PA[6]: 14203 2914 3356 29748 33437 AS_TRANS
  AS path: AS4 PA[2]: 33437 393219
  AS path: Merged[6]: 14203 2914 3356 29748 33437 393219 I
  Communities: 2914:420
```

**show route
receive-protocol bgp**

```
user@host> show route receive-protocol bgp 10.0.0.9 logical-system PE4 extensive

inet.0: 12 destinations, 13 routes (12 active, 0 holddown, 0 hidden)
* 10.0.0.0/30 (1 entry, 1 announced)
```

**logical-system
extensive**

```
Accepted
Route Label: 3
Nexthop: 10.0.0.9
AS path: 13979 I

* 10.0.0.4/30 (1 entry, 1 announced)
Accepted
Route Label: 3
Nexthop: 10.0.0.9
AS path: 13979 I

10.0.0.8/30 (2 entries, 1 announced)
Accepted
Route Label: 3
Nexthop: 10.0.0.9
AS path: 13979 I

* 10.9.9.1/32 (1 entry, 1 announced)
Accepted
Route Label: 3
Nexthop: 10.0.0.9
AS path: 13979 I

* 10.100.1.1/32 (1 entry, 1 announced)
Accepted
Route Label: 3
Nexthop: 10.0.0.9
AS path: 13979 I

* 44.0.0.0/24 (1 entry, 1 announced)
Accepted
Route Label: 300096
Nexthop: 10.0.0.9
AS path: 13979 I
AIGP: 203

* 55.0.0.0/24 (1 entry, 1 announced)
Accepted
Route Label: 300112
Nexthop: 10.0.0.9
AS path: 13979 7018 I
AIGP: 25

* 66.0.0.0/24 (1 entry, 1 announced)
Accepted
Route Label: 300144
Nexthop: 10.0.0.9
AS path: 13979 7018 I

* 99.0.0.0/24 (1 entry, 1 announced)
Accepted
Route Label: 300160
Nexthop: 10.0.0.9
AS path: 13979 7018 I
```

**show route
receive-protocol bgp
detail (Layer 2 VPN)**

```
user@host> show route receive-protocol bgp 10.255.14.171 detail
inet.0: 68 destinations, 68 routes (67 active, 0 holddown, 1 hidden)
Prefix                Nexthop                MED    Lclpref AS path
inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Prefix                Nexthop                MED    Lclpref AS path
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```

Prefix          Nexthop          MED    Lclpref AS path
mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED    Lclpref AS path
frame-vpn.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
Prefix          Nexthop          MED    Lclpref AS path
10.255.245.35:1:5:1/96 (1 entry, 1 announced)
  Route Distinguisher: 10.255.245.35:1
  Label-base : 800000, range : 4, status-vector : 0x0
  Nexthop: 10.255.245.35
  Localpref: 100
  AS path: I
  Communities: target:65299:100 Layer2-info: encaps:FRAME RELAY,
control flags: 0, mtu: 0
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED    Lclpref AS path
10.255.245.35:1:5:1/96 (1 entry, 0 announced)
  Route Distinguisher: 10.255.245.35:1
  Label-base : 800000, range : 4, status-vector : 0x0
  Nexthop: 10.255.245.35
  Localpref: 100
  AS path: I
  Communities: target:65299:100 Layer2-info: encaps:FRAME RELAY,
control flags:0, mtu: 0

```

show route
receive-protocol bgp

```

user@host> show route receive-protocol bgp 10.255.14.171 extensive
inet.0: 68 destinations, 68 routes (67 active, 0 holddown, 1 hidden)
Prefix          Nexthop          MED    Lclpref AS path

```

extensive (Layer 2 VPN)

```

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
frame-vpn.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
10.255.245.35:1:5:1/96 (1 entry, 1 announced)
  Route Distinguisher: 10.255.245.35:1
  Label-base : 800000, range : 4, status-vector : 0x0
  Nexthop: 10.255.245.35
  Localpref: 100
  AS path: I
  Communities: target:65299:100 Layer2-info: encaps:FRAME RELAY,
  control flags:0, mtu: 0
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
10.255.245.35:1:5:1/96 (1 entry, 0 announced)
  Route Distinguisher: 10.255.245.35:1
  Label-base : 800000, range : 4, status-vector : 0x0
  Nexthop: 10.255.245.35
  Localpref: 100
  AS path: I
  Communities: target:65299:100 Layer2-info: encaps:FRAME RELAY,
  control flags:0, mtu: 0

```

**show route
receive-protocol bgp
(Layer 3 VPN)**

```

user@host> show route receive-protocol bgp 10.255.14.171
inet.0: 33 destinations, 33 routes (32 active, 0 holddown, 1 hidden)
Prefix          Nexthop          MED      Lclpref AS path
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
VPN-A.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
10.255.14.175/32  10.255.14.171          100 2 I
10.255.14.179/32  10.255.14.171          2    100 I
VPN-B.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
10.255.14.175/32  10.255.14.171          100 2 I
10.255.14.177/32  10.255.14.171          100 I
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
mpls.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
bgp.l3vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
Prefix          Nexthop          MED      Lclpref AS path
10.255.14.171:300:10.255.14.177/32
                  10.255.14.171          100 I
10.255.14.171:100:10.255.14.179/32
                  10.255.14.171          2    100 I
10.255.14.171:200:10.255.14.175/32
                  10.255.14.171          100 2 I

```

**show route
receive-protocol bgp
detail (Layer 3 VPN)**

```

user@host> show route receive-protocol bgp 10.255.14.174 detail
inet.0: 16 destinations, 17 routes (15 active, 0 holddown, 1 hidden)
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
vpna.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
* 10.49.0.0/30 (1 entry, 1 announced)
  Route Distinguisher: 10.255.14.176:2
  VPN Label: 101264

```



```

    Nexthop: 10.255.14.174
    Localpref: 100
    AS path: I
    Communities: target:200:100
    AttrSet AS: 100
        Localpref: 100
        AS path: I
* 10.255.14.172/32 (1 entry, 1 announced)
    Route Distinguisher: 10.255.14.176:2
    VPN Label: 101280
    Nexthop: 10.255.14.174
    Localpref: 100
    AS path: I
    Communities: target:200:100
    AttrSet AS: 100
        Localpref: 100
        AS path: I
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
* 10.255.14.174:2:10.49.0.0/30 (1 entry, 0 announced)
    Route Distinguisher: 10.255.14.174:2
    VPN Label: 101264
    Nexthop: 10.255.14.174
    Localpref: 100
    AS path: I
    Communities: target:200:100
    AttrSet AS: 100
        Localpref: 100
        AS path: I
* 10.255.14.174:2:10.255.14.172/32 (1 entry, 0 announced)
    Route Distinguisher: 10.255.14.174:2
    VPN Label: 101280
    Nexthop: 10.255.14.174
    Localpref: 100
    AS path: I
    Communities: target:200:100
    AttrSet AS: 100
        Localpref: 100
        AS path: I
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

```

show route
receive-protocol bgp

```

user@host> show route receive-protocol bgp 10.255.245.63 extensive
inet.0: 244 destinations, 244 routes (243 active, 0 holddown, 1 hidden)
  Prefix                Nexthop                MED    Lclpref AS path

```

**extensive (Layer 3
VPN)**

```
1.1.1.0/24 (1 entry, 1 announced)
  Nexthop: 10.0.50.3
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.45
165.3.0.0/16 (1 entry, 1 announced)
  Nexthop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.68
165.4.0.0/16 (1 entry, 1 announced)
  Nexthop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.45
195.1.2.0/24 (1 entry, 1 announced)
  Nexthop: 111.222.5.254
  Localpref: 100
  AS path: I <Originator>
  Cluster list: 10.2.3.1
  Originator ID: 10.255.245.68
inet.2: 63 destinations, 63 routes (63 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED    Lclpref AS path
inet.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED    Lclpref AS path
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED    Lclpref AS path
mpls.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
```

show route table

Syntax	show route table <i>routing-table-name</i> <brief detail extensive terse> <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches)	show route table <i>routing-table-name</i> <brief detail extensive terse>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display the route entries in a particular routing table.
Options	<p>brief detail extensive terse—(Optional) Display the specified level of output.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>routing-table-name</i>—Display route entries for all routing tables whose name begins with this string (for example, inet.0 and inet6.0 are both displayed when you run the show route table inet command).</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show route summary
List of Sample Output	show route table bgp.l2.vpn on page 277 show route table bgp.l3vpn.0 on page 277 show route table bgp.l3vpn.0 detail on page 277 show route table bgp.rtarget.0 (When Proxy BGP Route Target Filtering Is Configured) on page 278 show route table inet.0 on page 279 show route table inet6.0 on page 279 show route table inet6.3 on page 279 show route table inetflow detail on page 279 show route table l2circuit.0 on page 280 show route table mpls on page 280 show route table mpls extensive on page 281 show route table mpls.0 on page 281 show route table mpls.0 (RSVP Route—Transit LSP) on page 281 show route table vpls_1 detail on page 282 show route table vpn-a on page 282 show route table vpn-a.mdt.0 on page 282 show route table VPN-A detail on page 283 show route table VPN-AB.inet.0 on page 283 show route table VPN_blue.mvpn-inet6.0 on page 283 show route table VPN-A detail on page 284

[show route table inetflow detail on page 284](#)

Output Fields For information about output fields, see the output field tables for the [show route](#) command, the [show route detail](#) command, the [show route extensive](#) command, or the [show route terse](#) command.

Sample Output

show route table bgp.l2vpn

```
user@host> show route table bgp.l2vpn
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.24.1:1:4:1/96
    *[BGP/170] 01:08:58, localpref 100, from 192.168.24.1
    AS path: I
    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am
```

show route table bgp.l3vpn.0

```
user@host> show route table bgp.l3vpn.0
bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.71.15:100:10.255.71.17/32
    *[BGP/170] 00:03:59, MED 1, localpref 100, from
10.255.71.15
    AS path: I
    > via so-2/1/0.0, Push 100020, Push 100011(top)
10.255.71.15:200:10.255.71.18/32
    *[BGP/170] 00:03:59, MED 1, localpref 100, from
10.255.71.15
    AS path: I
    > via so-2/1/0.0, Push 100021, Push 100011(top)
```

show route table bgp.l3vpn.0 detail

```
user@host> show route table bgp.l3vpn.0 detail
bgp.l3vpn.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

10.255.245.12:1:4.0.0.0/8 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 10.255.245.12:1
    Source: 10.255.245.12
    Next hop: 192.168.208.66 via fe-0/0/0.0, selected
    Label operation: Push 182449
    Protocol next hop: 10.255.245.12
    Push 182449
    Indirect next hop: 863a630 297
    State: <Active Int Ext>
    Local AS: 35 Peer AS: 35
    Age: 12:19 Metric2: 1
    Task: BGP_35.10.255.245.12+179
    Announcement bits (1): 0-BGP.0.0.0.0+179
    AS path: 30 10458 14203 2914 3356 I (Atomic) Aggregator: 3356 4.68.0.11

    Communities: 2914:420 target:11111:1 origin:56:78
    VPN Label: 182449
    Localpref: 100
    Router ID: 10.255.245.12

10.255.245.12:1:4.17.225.0/24 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 10.255.245.12:1
    Source: 10.255.245.12
    Next hop: 192.168.208.66 via fe-0/0/0.0, selected
    Label operation: Push 182465
    Protocol next hop: 10.255.245.12
    Push 182465
```

```

Indirect next hop: 863a8f0 305
State: <Active Int Ext>
Local AS: 35 Peer AS: 35
Age: 12:19 Metric2: 1
Task: BGP_35.10.255.245.12+179
Announcement bits (1): 0-BGP.0.0.0.0+179
AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496 I
Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
VPN Label: 182465
Localpref: 100
Router ID: 10.255.245.12

10.255.245.12:1:4.17.226.0/23 (1 entry, 1 announced)
*BGP Preference: 170/-101
Route Distinguisher: 10.255.245.12:1
Source: 10.255.245.12
Next hop: 192.168.208.66 via fe-0/0/0.0, selected
Label operation: Push 182465
Protocol next hop: 10.255.245.12
Push 182465
Indirect next hop: 86bd210 330
State: <Active Int Ext>
Local AS: 35 Peer AS: 35
Age: 12:19 Metric2: 1
Task: BGP_35.10.255.245.12+179
Announcement bits (1): 0-BGP.0.0.0.0+179
AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496
6496 I
Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
VPN Label: 182465
Localpref: 100
Router ID: 10.255.245.12

10.255.245.12:1:4.17.251.0/24 (1 entry, 1 announced)
*BGP Preference: 170/-101
Route Distinguisher: 10.255.245.12:1
Source: 10.255.245.12
Next hop: 192.168.208.66 via fe-0/0/0.0, selected
Label operation: Push 182465
Protocol next hop: 10.255.245.12
Push 182465
Indirect next hop: 86bd210 330
State: <Active Int Ext>
Local AS: 35 Peer AS: 35
Age: 12:19 Metric2: 1
Task: BGP_35.10.255.245.12+179
Announcement bits (1): 0-BGP.0.0.0.0+179
AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496
6496 I
Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
VPN Label: 182465
Localpref: 100

```

**show route table
bgp.rtarget.0 (When
Proxy BGP Route**

```

user@host> show route table bgp.rtarget.0
bgp.rtarget.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

Target Filtering Is Configured)

```
100:100:100/96
*[RTarget/5] 00:03:14
  Type Proxy
    for 10.255.165.103
    for 10.255.166.124
  Local
```

show route table inet.0

```
user@host> show route table inet.0
inet.0: 12 destinations, 12 routes (11 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:51:57
                   > to 111.222.5.254 via fxp0.0
1.0.0.1/32         *[Direct/0] 00:51:58
                   > via at-5/3/0.0
1.0.0.2/32         *[Local/0] 00:51:58
                   Local
12.12.12.21/32     *[Local/0] 00:51:57
                   Reject
13.13.13.13/32     *[Direct/0] 00:51:58
                   > via t3-5/2/1.0
13.13.13.14/32     *[Local/0] 00:51:58
                   Local
13.13.13.21/32     *[Local/0] 00:51:58
                   Local
13.13.13.22/32     *[Direct/0] 00:33:59
                   > via t3-5/2/0.0
127.0.0.1/32      [Direct/0] 00:51:58
                   > via lo0.0
111.222.5.0/24     *[Direct/0] 00:51:58
                   > via fxp0.0
111.222.5.81/32   *[Local/0] 00:51:58
                   Local
```

show route table inet6.0

```
user@host> show route table inet6.0
inet6.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Route, * = Both

fec0:0:0:3::/64   *[Direct/0] 00:01:34
>via fe-0/1/0.0

fec0:0:0:3::/128  *[Local/0] 00:01:34
>Local

fec0:0:0:4::/64   *[Static/5] 00:01:34
>to fec0:0:0:3::ffff via fe-0/1/0.0
```

show route table inet6.3

```
user@router> show route table inet6.3
inet6.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

::10.255.245.195/128
                   *[LDP/9] 00:00:22, metric 1
                   > via so-1/0/0.0
::10.255.245.196/128
                   *[LDP/9] 00:00:08, metric 1
                   > via so-1/0/0.0, Push 100008
```

show route table inetflow detail

```

user@host> show route table inetflow detail
inetflow.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.12.44.1,*/48 (1 entry, 1 announced)
    *BGP      Preference: 170/-101
                Next-hop reference count: 2
                State: **Active Ext>
                Local AS: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 I
                Communities: traffic-rate:0:0
                Validation state: Accept, Originator: 10.12.99.5
                Via: 10.12.44.0/24, Active
                Localpref: 100
                Router ID: 10.255.71.161

10.12.56.1,*/48 (1 entry, 1 announced)
    *Flow      Preference: 5
                Next-hop reference count: 2
                State: **Active>
                Local AS: 65002
                Age: 6:30
                Task: RT Flow
                Announcement bits (2): 0-Flow 1-BGP.0.0.0.0+179
                AS path: I
                Communities: 1:1

```

show route table l2circuit.0

```

user@host> show route table l2circuit.0
l2circuit.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.195:NoCtrlWord:1:1:Local/96
    * [L2CKT/7] 00:50:47
        > via so-0/1/2.0, Push 100049
        via so-0/1/3.0, Push 100049
10.1.1.195:NoCtrlWord:1:1:Remote/96
    * [LDP/9] 00:50:14
        Discard
10.1.1.195:CtrlWord:1:2:Local/96
    * [L2CKT/7] 00:50:47
        > via so-0/1/2.0, Push 100049
        via so-0/1/3.0, Push 100049
10.1.1.195:CtrlWord:1:2:Remote/96
    * [LDP/9] 00:50:14
        Discard

```

show route table mpls

```

user@host> show route table mpls
mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0          * [MPLS/0] 00:13:55, metric 1
            Receive
1          * [MPLS/0] 00:13:55, metric 1
            Receive
2          * [MPLS/0] 00:13:55, metric 1
            Receive
1024       * [VPN/0] 00:04:18
            to table red.inet.0, Pop

```


show route table mpls extensive

```

user@host> show route table mpls extensive
100000 (1 entry, 1 announced)
TSI:
KRT in-kernel 100000 /36 -> {so-1/0/0.0}
    *LDP    Preference: 9
            Next hop: via so-1/0/0.0, selected
            Pop
            State: <Active Int>
            Age: 29:50      Metric: 1
            Task: LDP
            Announcement bits (1): 0-KRT
            AS path: I
            Prefixes bound to route: 10.0.0.194/32

```

show route table mpls.0

```

user@host> show route table mpls.0
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0          *[MPLS/0] 00:45:09, metric 1
            Receive
1          *[MPLS/0] 00:45:09, metric 1
            Receive
2          *[MPLS/0] 00:45:09, metric 1
            Receive
100000     *[L2VPN/7] 00:43:04
            > via so-0/1/0.1, Pop
100001     *[L2VPN/7] 00:43:03
            > via so-0/1/0.2, Pop      Offset: 4
100002     *[LDP/9] 00:43:22, metric 1
            via so-0/1/2.0, Pop
            > via so-0/1/3.0, Pop
100002(S=0) *[LDP/9] 00:43:22, metric 1
            via so-0/1/2.0, Pop
            > via so-0/1/3.0, Pop
100003     *[LDP/9] 00:43:22, metric 1
            > via so-0/1/2.0, Swap 100002
            via so-0/1/3.0, Swap 100002
100004     *[LDP/9] 00:43:16, metric 1
            via so-0/1/2.0, Swap 100049
            > via so-0/1/3.0, Swap 100049
so-0/1/0.1 *[L2VPN/7] 00:43:04
            > via so-0/1/2.0, Push 100001, Push 100049(top)
            via so-0/1/3.0, Push 100001, Push 100049(top)
so-0/1/0.2 *[L2VPN/7] 00:43:03
            via so-0/1/2.0, Push 100000, Push 100049(top) Offset: -4
            > via so-0/1/3.0, Push 100000, Push 100049(top) Offset: -4

```

show route table mpls.0 (RSVP Route—Transit LSP)

```

user@host> show route table mpls.0
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0          *[MPLS/0] 00:37:31, metric 1
            Receive
1          *[MPLS/0] 00:37:31, metric 1
            Receive
2          *[MPLS/0] 00:37:31, metric 1
            Receive

```

```

13          *[MPLS/0] 00:37:31, metric 1
            Receive
300352      *[RSVP/7/1] 00:08:00, metric 1
            > to 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300352(S=0) *[RSVP/7/1] 00:08:00, metric 1
            > to 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300384      *[RSVP/7/2] 00:05:20, metric 1
            > to 8.64.1.106 via ge-1/0/0.0, Pop
300384(S=0) *[RSVP/7/2] 00:05:20, metric 1
            > to 8.64.1.106 via ge-1/0/0.0, Pop

```

show route table vpls_1 detail

```

user@host> show route table vpls_1 detail
vpls_1.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

1.1.1.11:1000:1:1/96 (1 entry, 1 announced)
*L2VPN Preference: 170/-1
Receive table: vpls_1.l2vpn.0
Next-hop reference count: 2
State: <Active Int Ext>
Age: 4:29:47 Metric2: 1
Task: vpls_1-l2vpn
Announcement bits (1): 1-BGP.0.0.0.0+179
AS path: I
Communities: Layer2-info: encaps:VPLS, control flags:Site-Down
Label-base: 800000, range: 8, status-vector: 0xFF

```

show route table vpn-a

```

user@host> show route table vpn-a
vpn-a.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both
192.168.16.1:1:1/96
    *[VPN/7] 05:48:27
    Discard
192.168.24.1:1:2:1/96
    *[BGP/170] 00:02:53, localpref 100, from 192.168.24.1
    AS path: I
    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am
192.168.24.1:1:3:1/96
    *[BGP/170] 00:02:53, localpref 100, from 192.168.24.1
    AS path: I
    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am

```

show route table vpn-a.mdt.0

```

user@host> show route table vpn-a.mdt.0
vpn-a.mdt.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1:0:10.255.14.216:232.1.1.1/144
    *[MVPN/70] 01:23:05, metric2 1
    Indirect
1:1:1:10.255.14.218:232.1.1.1/144
    *[BGP/170] 00:57:49, localpref 100, from 10.255.14.218
    AS path: I
    > via so-0/0/0.0, label-switched-path r0e-to-r1
1:1:2:10.255.14.217:232.1.1.1/144
    *[BGP/170] 00:57:49, localpref 100, from 10.255.14.217
    AS path: I
    > via so-0/0/1.0, label-switched-path r0-to-r2

```

show route table VPN-A detail

```

user@host> show route table VPN-A detail
VPN-AB.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
10.255.179.9/32 (1 entry, 1 announced)
    *BGP      Preference: 170/-101
                Route Distinguisher: 10.255.179.13:200
                Next hop type: Indirect
                Next-hop reference count: 5
                Source: 10.255.179.13
                Next hop type: Router, Next hop index: 732
                Next hop: 10.39.1.14 via fe-0/3/0.0, selected
                Label operation: Push 299824, Push 299824(top)
                Protocol next hop: 10.255.179.13
                Push 299824
                Indirect next hop: 8f275a0 1048574
                State: (Secondary Active Int Ext)
                Local AS: 1 Peer AS: 1
                Age: 3:41:06 Metric: 1 Metric2: 1
                Task: BGP_1.10.255.179.13+64309
                Announcement bits (2): 0-KRT 1-BGP RT Background
                AS path: I
                Communities: target:1:200 rte-type:0.0.0.0:1:0
                Import Accepted
                VPN Label: 299824 TTL Action: vrf-ttl-propagate
                Localpref: 100
                Router ID: 10.255.179.13
                Primary Routing Table bgp.l3vpn.0

```

show route table VPN-AB.inet.0

```

user@host> show route table VPN-AB.inet.0
VPN-AB.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.39.1.0/30      *[OSPF/10] 00:07:24, metric 1
                  > via so-7/3/1.0
10.39.1.4/30      *[Direct/0] 00:08:42
                  > via so-5/1/0.0
10.39.1.6/32      *[Local/0] 00:08:46
                  Local
10.255.71.16/32   *[Static/5] 00:07:24
                  > via so-2/0/0.0
10.255.71.17/32   *[BGP/170] 00:07:24, MED 1, localpref 100, from
10.255.71.15
                  AS path: I
                  > via so-2/1/0.0, Push 100020, Push 100011(top)
10.255.71.18/32   *[BGP/170] 00:07:24, MED 1, localpref 100, from
10.255.71.15
                  AS path: I
                  > via so-2/1/0.0, Push 100021, Push 100011(top)
10.255.245.245/32 *[BGP/170] 00:08:35, localpref 100
                  AS path: 2 I
                  > to 10.39.1.5 via so-5/1/0.0
10.255.245.246/32 *[OSPF/10] 00:07:24, metric 1
                  > via so-7/3/1.0

```

show route table VPN_blue.mvpn-inet6.0

```

user@host> show route table VPN_blue.mvpn-inet6.0
vpn_blue.mvpn-inet6.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:10.255.2.202:65535:10.255.2.202/432

```

```

* [BGP/170] 00:02:37, localpref 100, from 10.255.2.202
  AS path: I
  > via so-0/1/3.0
1:10.255.2.203:65535:10.255.2.203/432
* [BGP/170] 00:02:37, localpref 100, from 10.255.2.203
  AS path: I
  > via so-0/1/0.0
1:10.255.2.204:65535:10.255.2.204/432
* [MVPN/70] 00:57:23, metric2 1
  Indirect
5:10.255.2.202:65535:128::192.168.90.2:128:ffff::1/432
* [BGP/170] 00:02:37, localpref 100, from 10.255.2.202
  AS path: I
  > via so-0/1/3.0
6:10.255.2.203:65535:65000:128::10.12.53.12:128:ffff::1/432
* [PIM/105] 00:02:37
  Multicast (IPv6)
7:10.255.2.202:65535:65000:128::192.168.90.2:128:ffff::1/432
* [MVPN/70] 00:02:37, metric2 1
  Indirect

```

show route table VPN-A detail

```

user@host> show route table VPN-A detail
VPN-AB.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
10.255.179.9/32 (1 entry, 1 announced)
  *BGP
    Preference: 170/-101
    Route Distinguisher: 10.255.179.13:200
    Next hop type: Indirect
    Next-hop reference count: 5
    Source: 10.255.179.13
    Next hop type: Router, Next hop index: 732
    Next hop: 10.39.1.14 via fe-0/3/0.0, selected
    Label operation: Push 299824, Push 299824(top)
    Protocol next hop: 10.255.179.13
    Push 299824
    Indirect next hop: 8f275a0 1048574
    State: (Secondary Active Int Ext)
    Local AS: 1 Peer AS: 1
    Age: 3:41:06 Metric: 1 Metric2: 1
    Task: BGP_1.10.255.179.13+64309
    Announcement bits (2): 0-KRT 1-BGP RT Background
    AS path: I
    Communities: target:1:200 rte-type:0.0.0.0:1:0
    Import Accepted
    VPN Label: 299824 TTL Action: vrf-ttl-propagate
    Localpref: 100
    Router ID: 10.255.179.13
    Primary Routing Table bgp.13vpn.0

```

show route table inetflow detail

```

user@host> show route table inetflow detail
inetflow.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.12.44.1,*/48 (1 entry, 1 announced)
  *BGP
    Preference: 170/-101
    Next-hop reference count: 2
    State: **Active Ext>
    Local AS: 65002 Peer AS: 65000
    Age: 4
    Task: BGP_65000.10.12.99.5+3792
    Announcement bits (1): 0-Flow
    AS path: 65000 I
    Communities: traffic-rate:0:0

```

```

Validation state: Accept, Originator: 10.12.99.5
Via: 10.12.44.0/24, Active
Localpref: 100
Router ID: 10.255.71.161

10.12.56.1,*/48 (1 entry, 1 announced)
  *Flow Preference: 5
    Next-hop reference count: 2
    State: **Active>
    Local AS: 65002
    Age: 6:30
    Task: RT Flow
    Announcement bits (2): 0-Flow 1-BGP.0.0.0.0+179
    AS path: I
    Communities: 1:1

user@PE1> show route table green.l2vpn.0 (VPLS Multihoming with FEC 129)
green.l2vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.2:100:1.1.1.2/96 AD
    *[VPLS/170] 1d 03:11:03, metric2 1
    Indirect
1.1.1.4:100:1.1.1.4/96 AD
    *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
    AS path: I, validation-state: unverified
    > via ge-1/2/1.5
1.1.1.2:100:1:0/96 MH
    *[VPLS/170] 1d 03:11:03, metric2 1
    Indirect
1.1.1.4:100:1:0/96 MH
    *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
    AS path: I, validation-state: unverified
    > via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.2:1.1.1.4/176
    *[VPLS/7] 1d 03:11:02, metric2 1
    > via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.4:1.1.1.2/176
    *[LDP/9] 1d 03:11:02
    Discard

```

show route terse


Syntax	show route terse <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches)	show route terse
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Display a high-level summary of the routes in the routing table.
	<div>  <p>NOTE: For BGP routes, the show route terse command displays the local preference attribute and MED instead of the metric1 and metric2 values. This is mostly due to historical reasons.</p> <p>To display the metric1 and metric2 value of a BGP route, use the show route extensive command.</p> </div>
Options	<p>none—Display a high-level summary of the routes in the routing table.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Required Privilege Level	view
List of Sample Output	show route terse on page 288
Output Fields	Table 22 on page 286 describes the output fields for the show route terse command. Output fields are listed in the approximate order in which they appear.

Table 22: show route terse Output Fields

Field Name	Field Description
<i>routing-table-name</i>	Name of the routing table (for example, inet.0).
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.
<i>number routes</i>	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> active (routes that are active) holddown (routes that are in the pending state before being declared inactive) hidden (routes that are not used because of a routing policy)

Table 22: show route terse Output Fields (*continued*)

Field Name	Field Description
route key	Key for the state of the route: <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • -—A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route.
A	Active route. An asterisk (*) indicates this is the active route.
V	Validation status of the route: <ul style="list-style-type: none"> • ?—Not evaluated. Indicates that the route was not learned through BGP. • I—Invalid. Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • N—Unknown. Indicates that the prefix is not among the prefixes or prefix ranges in the database. • V—Valid. Indicates that the prefix and autonomous system pair are found in the database.
Destination	Destination of the route.
P	Protocol through which the route was learned: <ul style="list-style-type: none"> • A—Aggregate • B—BGP • C—CCC • D—Direct • G—GMPLS • I—IS-IS • L—L2CKT, L2VPN, LDP, Local • K—Kernel • M—MPLS, MSDP • O—OSPF • P—PIM • R—RIP, RIPng • S—Static • T—Tunnel
Prf	Preference value of the route. In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value and a lower Preference2 value.
Metric 1	First metric value in the route. For routes learned from BGP, this is the MED metric.
Metric 2	Second metric value in the route. For routes learned from BGP, this is the IGP metric.

Table 22: show route terse Output Fields (*continued*)

Field Name	Field Description
Next hop	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.
AS path	<p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • ?—Incomplete; typically, the AS path was aggregated.

Sample Output

show route terse

```

user@host> show route terse
inet.0: 10 destinations, 12 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination      P Prf  Metric 1  Metric 2  Next hop      AS path
* ? 1.0.1.1/32        O 10      1          >10.0.0.2      I
?                               B 170      100          >10.0.0.2      I
  unverified                               >10.0.0.2
* ? 1.1.1.1/32        D 0          >10.0.0.2
* V 2.2.0.2/32        B 170      110      >10.0.0.2      200 I
  valid                               >10.0.0.2
* ? 10.0.0.0/30       D 0          >10.0.0.2
?                               B 170      100      >10.0.0.2      I
  unverified                               >10.0.0.2
* ? 10.0.0.1/32       L 0          Local
* ? 10.0.0.4/30       B 170      100          >10.0.0.2      I
  unverified                               >10.0.0.2
* ? 10.0.0.8/30       B 170      100          >10.0.0.2      I
  unverified                               >10.0.0.2
* I 172.16.1.1/32     B 170      90          >10.0.0.2      200 I
  invalid                               >10.0.0.2
* N 192.168.2.3/32    B 170      100          >10.0.0.2      200 I
  unknown                               >10.0.0.2
* ? 224.0.0.5/32      O 10      1          MultiRecv

```


test policy

Syntax	<code>test policy <i>policy-name</i> <i>prefix</i></code>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	Test a policy configuration to determine which prefixes match routes in the routing table.
Options	<i>policy-name</i> —Name of a policy. <i>prefix</i> —Destination prefix to match.
Additional Information	All prefixes in the default unicast routing table (inet.0) that match prefixes that are the same as or longer than the specific prefix are processed by the from clause in the specified policy. All prefixes accepted by the policy are displayed. The test policy command evaluates a policy differently from the BGP import process. When testing a policy that contains an interface match condition in the from clause, the test policy command uses the match condition. In contrast, BGP does not use the interface match condition when evaluating the policy against routes learned from internal BGP (IBGP) or external BGP (EGBP) multihop peers.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show policy damping
List of Sample Output	test policy on page 290
Output Fields	For information about output fields, see the output field tables for the show route command, the show route detail command, the show route extensive command, or the show route terse command.

Sample Output

test policy

```
user@host> test policy test-statics 3.0.0.1/8
inet.0: 44 destinations, 44 routes (44 active, 0 holddown, 0 hidden)
Prefixes passing policy:

3.0.0.0/8      *[BGP/170] 16:22:46, localpref 100, from 10.255.255.41
               AS Path: 50888 I
               > to 10.11.4.32 via en0.2, label-switched-path l2
3.3.3.1/32     *[IS-IS/18] 2d 00:21:46, metric 0, tag 2
               > to 10.0.4.7 via fxp0.0
3.3.3.2/32     *[IS-IS/18] 2d 00:21:46, metric 0, tag 2
               > to 10.0.4.7 via fxp0.0
3.3.3.3/32     *[IS-IS/18] 2d 00:21:46, metric 0, tag 2
               > to 10.0.4.7 via fxp0.0
3.3.3.4/32     *[IS-IS/18] 2d 00:21:46, metric 0, tag 2
               > to 10.0.4.7 via fxp0.0
Policy test-statics: 5 prefixes accepted, 0 prefixes rejected
```

PART 4

Troubleshooting

- [Routing Protocol Process Memory FAQs on page 293](#)
- [RIP Monitoring on page 301](#)

CHAPTER 6

Routing Protocol Process Memory FAQs

- [Routing Protocol Process Memory FAQs Overview on page 293](#)
- [Routing Protocol Process Memory FAQs on page 294](#)

Routing Protocol Process Memory FAQs Overview

Junos OS is based on the FreeBSD Unix operating system. The open source software is modified and hardened to operate in the device's specialized environment. For example, some executables have been deleted, while other utilities were de-emphasized. Additionally, certain software processes were added to enhance the routing functionality. The result of this transformation is the kernel, the heart of the Junos OS software.

The kernel is responsible for operating multiple processes that perform the actual functions of the device. Each process operates in its own protected memory space, while the communication among all the processes is still controlled by the kernel. This separation provides isolation between the processes, and resiliency in the event of a process failure. This is important in a core routing platform because a single process failure does not cause the entire device to cease functioning.

Some of the common software processes include the routing protocol process (rpd) that controls the device's protocols, the device control process (dcd) that controls the device's interfaces, the management process (mgd) that controls user access to the device, the chassis process (chassisd) that controls the device's properties itself, and the Packet Forwarding Engine process (pfed) that controls the communication between the device's Packet Forwarding Engine and the Routing Engine. The kernel also generates specialized processes as needed for additional functionality, such as SNMP, the Virtual Router Redundancy Protocol (VRRP), and Class of Service (CoS).

The routing protocol process is a software process within the Routing Engine software, which controls the routing protocols that run on the device. Its functionality includes all protocol messages, routing table updates, and implementation of routing policies.

The routing protocol process starts all configured routing protocols and handles all routing messages. It maintains one or more routing tables, which consolidate the routing information learned from all routing protocols. From this routing information, the routing protocol process determines the active routes to network destinations and installs these routes into the Routing Engine's forwarding table. Finally, it implements routing policy, which allows you to control the routing information that is transferred between the routing

protocols and the routing table. Using routing policy, you can filter and limit the transfer of information as well as set properties associated with specific routes.

Related Documentation

- [Routing Protocol Process Memory FAQs on page 294](#)

Routing Protocol Process Memory FAQs

The following sections present the most frequently asked questions and answers related to the routing protocol process memory utilization, operation, interpretation of related command outputs, and troubleshooting the software process.

Frequently Asked Questions: Routing Protocol Process Memory

This section presents frequently asked questions and answers related to the memory usage of the routing protocol process.

Why does the routing protocol process use excessive memory?

The routing protocol process uses hundreds of megabytes of RAM in the Routing Engine to store information needed for the operation of routing and related protocols, such as BGP, OSPF, IS-IS, RSVP, LDP and MPLS. Such huge consumption of memory is common for the process, as the information it stores includes routes, next hops, interfaces, routing policies, labels, and label-switched paths (LSPs). Because access to the RAM memory is much faster than access to the hard disk, most of the routing protocol process information is stored in the RAM memory instead of using the hard disk space. This ensures that the performance of the routing protocol process is maximized.

How can I check the amount of memory the routing protocol process is using?

You can check routing protocol process memory usage by entering the **show system processes** and the **show task memory** Junos OS command-line interface (CLI) operational mode commands.

The **show system processes** command displays information about software processes that are running on the device and that have controlling terminals. The **show task memory** command displays memory utilization for routing protocol tasks on the Routing Engine.

You can check the routing protocol process memory usage by using the **show system processes** command with the **extensive** option. The **show task memory** command displays a report generated by the routing protocol process on its own memory usage. However, this report does not display all the memory used by the process. The value reported by the routing protocol process does not account for the memory used for the **TEXT** and **STACK** segments, or the memory used by the process's internal memory manager. Further, the Resident Set Size value includes shared library pages used by the routing protocol process.

For more information about checking the routing protocol process memory usage, see [Check Routing Protocol Process \(rpd\) Memory Usage](#).

For more information, see the **show system processes** command and the **show task memory** command.

I just deleted a large number of routes from the routing protocol process. Why is it still using so much memory?

The **show system processes extensive** command displays a **RES** value measured in kilobytes. This value represents the amount of program memory resident in the physical memory. This is also known as RSS or Resident Set Size. The **RES** value includes shared library pages used by the process. Any amount of memory freed by the process might still be considered part of the **RES** value. Generally, the kernel delays the migrating of memory out of the **Inact** queue into the **Cache** or **Free** list unless there is a memory shortage. This can lead to large discrepancies between the values reported by the routing protocol process and the kernel, even after the routing protocol process has freed a large amount of memory.

Frequently Asked Questions: Interpreting Routing Protocol Process-Related Command Outputs

This section presents frequently asked questions and answers about the routing protocol process-related Junos OS command-line interface (CLI) command outputs that are used to display the memory usage of the routing protocol process.

How do I interpret memory numbers displayed in the show system processes extensive command output?

The **show system processes extensive** command displays exhaustive system process information about software processes that are running on the device and have controlling terminals. This command is equivalent to the UNIX **top** command. However, the UNIX **top** command shows real-time memory usage, with the memory values constantly changing, while the **show system processes extensive** command provides a snapshot of memory usage in a given moment.

To check overall CPU and memory usage, enter the **show system processes extensive** command. Refer to [Table 23 on page 296](#) for information about the **show system processes extensive** commands output fields.

```
user@host> show system processes extensive
last pid: 544; load averages: 0.00, 0.00, 0.00 18:30:33
37 processes: 1 running, 36 sleeping

Mem: 25M Active, 3968K Inact, 19M Wired, 184K Cache, 8346K Buf, 202M Free
Swap: 528M Total, 64K Used, 528M Free

  PID USERNAME PRI NICE SIZE RES STATE TIME WCPU CPU COMMAND
    544 root    30  0  604K 768K RUN   0:00 0.00% 0.00% top
      3 root    28  0    0K  12K psleep 0:00 0.00% 0.00% vmdaemon
      4 root    28  0    0K  12K update 0:03 0.00% 0.00% update
    528 aviva    18  0  660K 948K pause  0:00 0.00% 0.00% tcsh
    204 root    18  0  300K 544K pause  0:00 0.00% 0.00% csh
    131 root    18  0  332K 532K pause  0:00 0.00% 0.00% cron
    186 root    18  0  196K  68K pause  0:00 0.00% 0.00% watchdog
     27 root    10  0  512M 16288K mfsidl 0:00 0.00% 0.00% mount_mfs
      1 root    10  0  620K 344K wait   0:00 0.00% 0.00% init
    304 root     3  0  884K 900K ttyin  0:00 0.00% 0.00% bash
    200 root     3  0  180K 540K ttyin  0:00 0.00% 0.00% getty
    203 root     3  0  180K 540K ttyin  0:00 0.00% 0.00% getty
    202 root     3  0  180K 540K ttyin  0:00 0.00% 0.00% getty
    201 root     3  0  180K 540K ttyin  0:00 0.00% 0.00% getty
    194 root     2  0 2248K 1640K select 0:11 0.00% 0.00% rpd
    205 root     2  0   964K  800K select 0:12 0.00% 0.00% tnp.chassisd
```

```

189 root      2 -12  352K  740K select  0:03  0.00%  0.00% xntpd
114 root      2  0   296K  612K select  0:00  0.00%  0.00% amd
188 root      2  0   780K  600K select  0:00  0.00%  0.00% dcd
527 root      2  0   176K  580K select  0:00  0.00%  0.00% rlogind
195 root      2  0   212K  552K select  0:00  0.00%  0.00% inetd
187 root      2  0   192K  532K select  0:00  0.00%  0.00% tnetd
 83 root      2  0   188K  520K select  0:00  0.00%  0.00% syslogd
538 root      2  0  1324K  516K select  0:00  0.00%  0.00% mgd
 99 daemon    2  0   176K  492K select  0:00  0.00%  0.00% portmap
163 root      2  0   572K  420K select  0:00  0.00%  0.00% nsrexecd
192 root      2  0   560K  400K select  0:10  0.00%  0.00% snmpd
191 root      2  0  1284K  376K select  0:00  0.00%  0.00% mgd
537 aviva     2  0   636K  364K select  0:00  0.00%  0.00% cli
193 root      2  0   312K  204K select  0:07  0.00%  0.00% mib2d
  5 root      2  0      0K   12K pfesel  0:00  0.00%  0.00% if_pfe
  2 root     -18  0      0K   12K psleep  0:00  0.00%  0.00% pagedaemon
  0 root     -18  0      0K    0K sched   0:00  0.00%  0.00% swapper

```

Table 23 on page 296 describes the output fields that represent the memory values for the **show system processes extensive** command. Output fields are listed in the approximate order in which they appear.

Table 23: show system processes extensive Output Fields

Field Name	Field Description
Mem	Information about physical and virtual memory allocation.
Active	Memory allocated and actively used by the program.
Inact	Memory allocated but not recently used or memory freed by the programs. Inactive memory remains mapped in the address space of one or more processes and, therefore, counts toward the RSS value of those processes.
Wired	Memory that is not eligible to be swapped, usually used for in-kernel memory structures and/or memory physically locked by a process.
Cache	Memory that is not associated with any program and does not need to be swapped before being reused.
Buf	Size of memory buffer used to hold data recently called from the disk.
Free	Memory that is not associated with any programs. Memory freed by a process can become Inactive , Cache , or Free , depending on the method used by the process to free the memory.
Swap	Information about swap memory. <ul style="list-style-type: none"> • Total—Total memory available to be swapped to disk. • Used—Memory swapped to disk. • Free—Memory available for further swap.

The rest of the command output displays information about the memory usage of each process. The **SIZE** field indicates the size of the virtual address space, and the **RES** field indicates the amount of the program in physical memory, which is also known as RSS or Resident Set Size. For more information, see the **show system processes** command.

What is the difference between Active and Inact memory that is displayed by the show system processes extensive command?

When the system is under memory pressure, the pageout process reuses memory from the free, cache, inactive and, if necessary, active pages. When the pageout process runs, it scans memory to see which pages are good candidates to be unmapped and freed up. Thus, the distinction between **Active** and **Inact** memory is only used by the pageout process to determine which pool of pages to free first at the time of a memory shortage.

The pageout process first scans the **Inact** list, and checks whether the pages on this list have been accessed since the time they have been listed here. The pages that have been accessed are moved from the **Inact** list to the **Active** list. On the other hand, pages that have not been accessed become prime candidates to be freed by the pageout process. If the pageout process cannot produce enough free pages from the **Inact** list, pages from the **Active** list get freed up.

Because the pageout process runs only when the system is under memory pressure, the pages on the **Inact** list remain untouched – even if they have not been accessed recently – when the amount of **Free** memory is adequate.

How do I interpret memory numbers displayed in the show task memory command output?

The **show task memory** command provides a comprehensive picture of the memory utilization for routing protocol tasks on the Routing Engine. The routing protocol process is the main task that uses Routing Engine memory.

To check routing process memory usage, enter the **show task memory** command. Refer to [Table 24 on page 297](#) for information about the **show task memory** command output fields.

```
user@host> show task memory
Memory          Size (kB)  %Available  When
Currently In Use:    29417      3%         now
Maximum Ever Used:   33882      4%         00/02/11 22:07:03
Available:          756281     100%        now
```

[Table 24 on page 297](#) describes the output fields for the **show task memory** command. Output fields are listed in the approximate order in which they appear.

Table 24: show task memory Output Fields

Field Name	Field Description
Memory Currently In Use	Memory currently in use. Dynamically allocated memory plus the DATA segment memory in kilobytes.
Memory Maximum Ever Used	Maximum memory ever used.
Memory Available	Memory currently available.

The **show task memory** command does not display all the memory used by the routing protocol process. This value does not account for the memory used for the **TEXT** and

STACK segments, or the memory used by the routing protocol process's internal memory manager.

Why is the Currently In Use value less than the RES value?

The **show task memory** command displays a **Currently In Use** value measured in kilobytes. This value represents the memory currently in use. It is the dynamically allocated memory plus the **DATA** segment memory. The **show system processes extensive** command displays a **RES** value measured in kilobytes. This value represents the amount of program memory resident in the physical memory. This is also known as RSS or Resident Set Size.

The **Currently In Use** value does not account for all of the memory that the routing protocol process uses. This value does not include the memory used for the **TEXT** and the **STACK** segments, and a small percentage of memory used by the routing protocol process's internal memory manager. Further, the **RES** value includes shared library pages used by the routing protocol process.

Any amount of memory freed by the routing protocol process might still be considered part of the **RES** value. Generally, the kernel delays the migrating of memory out of the **Inact** queue into the **Cache** or **Free** list unless there is a memory shortage. This can lead to large discrepancies between the **Currently In Use** value and the **RES** value.

Frequently Asked Questions: Routing Protocol Process Memory Swapping

This section presents frequently asked questions and answers related to the memory swapping of the routing protocol process from the Routing Engine memory to the hard disk memory.

How do I monitor swap activity?

When the system is under memory pressure, the pageout process reuses memory from the free, cache, inactive and, if necessary, active pages. You can monitor the swap activity by viewing the syslog message reported by the kernel during periods of high pageout activity.

The syslog message appears as follows:

```
Mar  3 20:08:02 olympic /kernel: High pageout rate!! 277 pages/sec.
```

You can use the **vmstat -s** command to print the statistics for the swapout activity. The displayed statistics appear as follows:

```
0 swap pager pageouts
0 swap pager pages paged out
```

The **swap pager pageouts** is the number of pageout operations to the swap device, and the **swap pager pages paged out** is the number of pages paged out to the swap device.

Why does the system start swapping when I try to dump core using the request system core-dumps command?

The **request system core-dumps** command displays a list of system core files created when the device has failed. This command can be useful for diagnostic purposes. Each list item includes the file permissions, number of links, owner, group, size, modification

date, path, and filename. You can use the **core-filename** option and the **core-file-info**, **brief**, and **detail** options to display more information about the specified core-dump files.

You can use the **request system core-dumps** command to perform a non-fatal core-dump without aborting the routing protocol process. To do this, the routing protocol process is forked, generating a second copy, and then aborted. This process can double the memory consumed by the two copies of the routing protocol processes, pushing the system into swap.

Why does the show system processes extensive command show that memory is swapped to disk although there is plenty of free memory?

Memory can remain swapped out indefinitely if it is not accessed again. Therefore, the **show system processes extensive** command shows that memory is swapped to disk even though there is plenty of free memory, and such a situation is not unusual.

Frequently Asked Questions: Troubleshooting the Routing Protocol Process

This section presents frequently asked questions and answers related to a shortage of memory and memory leakage by the routing protocol process.

What does the RPD_OS_MEMHIGH message mean?

The **RPD_OS_MEMHIGH** message is written into the system message file if the routing protocol process is running out of memory. This message alerts you that the routing protocol process is using the indicated amount and percentage of Routing Engine memory, which is considered excessive. This message is generated either because the routing protocol process is leaking memory or the use of system resources is excessive, perhaps because routing filters are misconfigured or the configured network topology is very complex.

When the memory utilization for the routing protocol process is using all available Routing Engine DRAM memory (Routing Engines with maximum 2 GB DRAM) or reaches the limit of 2 GB of memory (Routing Engines with 4 GB DRAM), a message of the following form is written every minute in the syslog message file:

RPD_OS_MEMHIGH: Using 188830 KB of memory, 100 percent of available

This message includes the amount, in kilobytes and/or the percentage, of the available memory in use.

This message should not appear under normal conditions, as any further memory allocations usually require a portion of existing memory to be written to swap. As a recommended solution, increase the amount of RAM in the Routing Engine. For more information, go to <http://kb.juniper.net/InfoCenter/index?page=content&id=KB14186>.

What can I do when there is a memory shortage even after a swap?

It is not recommended for the system to operate in this state, notwithstanding the existence of swap. The protocols that run in the routing protocol process usually have a real-time requirement that cannot reliably withstand the latency of being swapped to hard disk. If the memory shortage has not resulted from a memory leak, then either a

reduction in the memory usage or an upgrade to a higher memory-capacity Routing Engine is required.

How do I determine whether there is a memory leak in the routing protocol process?

Memory leaks are typically the result of a seemingly unbounded growth in the memory usage of a process as reported by the **show system processes extensive** command.

There are two classes of memory leaks that the routing protocol process can experience.

- The first class occurs when the allocated memory that is no longer in use is not freed. This class of leak can usually be fixed by taking several samples of the **show task memory detail** command over a period of time and comparing the deltas.
- The second class occurs when there is a late access to freed memory. If the access is not outside the mapped address space, the kernel backfills the accessed page with real memory. This backfill is done without the knowledge of the routing protocol process's internal memory allocator, which makes this class of leak much more difficult to resolve. If a memory leak of this class is suspected, writing the state of the system to a disk file (creating a core file) is suggested.

A large discrepancy between the **RES** value and the **Currently In Use** value might indicate a memory leak. However, large discrepancies can also occur for legitimate reasons. For example, the memory used for the **TEXT** and **STACK** segments or the memory used by the routing protocol process's internal memory manager might not be displayed. Further, the **RES** value includes shared library pages used by the process.

What is the task_timer?

The source of a routing protocol process memory leak can usually be identified by dumping the timers for each task. You can use the **show task task-name** command to display routing protocol tasks on the Routing Engine. Tasks can be baseline tasks performed regardless of the device's configuration, and other tasks that depend on the device configuration.

For more information, see the **show task** command.

Related Documentation

- [Routing Protocol Process Memory FAQs Overview on page 293](#)

CHAPTER 7

RIP Monitoring

- [Monitoring RIP Routing Information on page 301](#)

Monitoring RIP Routing Information

Purpose Use the monitoring functionality to monitor RIP routing on routing devices.

Action To view RIP routing information in the J-Web interface, select **Monitor > Routing > RIP Information**.

To view RIP routing information in the CLI, enter the following CLI commands:

- **show rip statistics**
- **show rip neighbor**

Meaning [Table 25 on page 301](#) summarizes key output fields in the RIP routing display in the J-Web interface.

Table 25: Summary of Key RIP Routing Output Fields

Field	Values	Additional Information
RIP Statistics		
Protocol Name	The RIP protocol name.	
Port number	The port on which RIP is enabled.	
Hold down time	The interval during which routes are neither advertised nor updated.	
Global routes learned	Number of RIP routes learned on the logical interface.	
Global routes held down	Number of RIP routes that are not advertised or updated during the hold-down interval.	
Global request dropped	Number of requests dropped.	

Table 25: Summary of Key RIP Routing Output Fields (*continued*)

Field	Values	Additional Information
Global responses dropped	Number of responses dropped.	
RIP Neighbors		
Neighbor	Name of the RIP neighbor.	This value is the name of the interface on which RIP is enabled. Click the name to see the details for this neighbor.
State	State of the RIP connection: Up or Dn (Down).	
Source Address	Local source address.	This value is the configured address of the interface on which RIP is enabled.
Destination Address	Destination address.	This value is the configured address of the immediate RIP adjacency.
Send Mode	The mode of sending RIP messages.	
Receive Mode	The mode in which messages are received.	
In Metric	Value of the incoming metric configured for the RIP neighbor.	

- Related Documentation**
- [Configuring a RIP Network \(J-Web Procedure\)](#)
 - [Layer 3 Protocols Supported on EX Series Switches](#)

PART 5

Index

- [Index on page 305](#)

Index

Symbols

#, comments in configuration statements.....	xiv
(), in syntax descriptions.....	xiv
< >, in syntax descriptions.....	xiv
[], in configuration statements.....	xiv
{ }, in configuration statements.....	xiv
(pipe), in syntax descriptions.....	xiv

A

advertisements, displaying	170
any-sender statement	
RIP.....	101
auth (tracing flag).....	128
authentication	
RIPv2, MD5.....	25
RIPv2, plain-text passwords.....	25
simple	
RIP.....	20
authentication configuration	
BFD.....	34
authentication-key statement	
RIP.....	102
usage guidelines.....	20
authentication-type statement	
RIP.....	103
usage guidelines.....	20

B

best routes, displaying.....	177
BFD	
authentication configuration.....	34
protocol.....	26, 27
bfd-liveness-detection statement	
RIP.....	104
usage guidelines.....	27
braces, in configuration statements.....	xiv
brackets	
angle, in syntax descriptions.....	xiv
square, in configuration statements.....	xiv

C

check-zero statement.....	107
clear rip general-statistics command.....	136
clear rip statistics command.....	137
comments, in configuration statements.....	xiv
conditions	
routing policy.....	150
connectivity	
unidirectional (RIP).....	7
conventions	
text and syntax.....	xiii
curly braces, in configuration statements.....	xiv
customer support.....	xv
contacting JTAC.....	xv

D

demand-circuit statement	
RIP.....	108
usage guidelines.....	77
diagnosis	
verifying RIP host reachability	19, 96
verifying RIP message exchange.....	94
verifying RIP-enabled interfaces.....	96
distance-vector routing protocols.....	3
See also RIP	
documentation	
comments on.....	xv

E

error (tracing flag)	
RIP.....	128
export route information, displaying.....	199
export statement	
RIP.....	109
usage guidelines.....	14

F

FAQs	
routing protocol process memory.....	293, 294
font conventions.....	xiii
forwarding table	
route entries, displaying.....	216

G

graceful-restart statement	
RIP.....	110
group statement	
RIP.....	111

groups	
RIP routers.....	14

H

hidden routes, displaying.....	230
holddown (tracing flag).....	128
holddown statement	
RIP.....	113
usage guidelines.....	70
hop count, maximizing.....	6
<i>See also</i> RIP	
host reachability	
verifying for RIP network hosts.....	19, 96

I

import statement	
RIP.....	114
usage guidelines.....	47
incoming metric (RIP)	
description.....	52

M

manuals	
comments on.....	xv
max-retrans-time statement.....	116
maximum hop count, RIP.....	6
message-size statement.....	117
metric-in statement	
RIP.....	118
usage guidelines.....	56
metric-out statement	
RIP.....	119
metrics	
RIP.....	119
minimum-interval statement	
RIP.....	104
usage guidelines.....	27
minimum-receive-interval statement	
RIP.....	104
usage guidelines.....	27
monitoring	
RIP.....	301
multiplier statement	
RIP.....	104
usage guidelines.....	27

N

neighbor statement	
RIP.....	120

neighbors <i>See</i> adjacencies, IS-IS; BGP peers; OSPF	
neighbors; RIP neighbors	
network interfaces	
verifying RIP message exchange.....	94
verifying RIP on.....	96
networks	
sample distance-vector routing.....	4
sample poison reverse routing.....	7
sample split horizon routing.....	6
sample unidirectional routing.....	7
next hops	
routes sent to, displaying.....	244
no-adaptation statement	
RIP.....	104
usage guidelines.....	27
no-check-zero statement.....	107

O

outgoing metric (RIP)	
description.....	52

P

packets (tracing flag)	
RIP.....	128
parentheses, in syntax descriptions.....	xiv
password	
for RIPv2 authentication.....	25
path cost metrics	
for RIP routes, description.....	52
for RIP routes, modifying.....	53, 55
poison reverse technique.....	6
policy <i>See</i> routing policies	
policy, routing	
RIP.....	109, 114
preference statement	
RIP.....	122
processes	
restarting.....	138
protocols	
distance vector <i>See</i> RIP	
RIP <i>See</i> RIP	
RIP, monitoring.....	301

R

reachability	
verifying for a RIP network.....	19, 96
receive statement	
RIP.....	123
usage guidelines.....	61

-
- restart command.....138
 - restarting
 - software processes.....138
 - rib-group statement
 - RIP.....124
 - usage guidelines.....65
 - RIP
 - authentication.....20, 102
 - authentication (RIPv2 only).....20
 - basic network (configuration editor).....14
 - BFD.....26, 27
 - demand circuit retransmission timer.....116
 - demand circuits
 - overview.....77
 - packets.....78
 - disabling address checks.....101
 - distance vector protocol.....3
 - efficiency techniques.....6
 - enabling.....124
 - general statistics
 - clearing.....136
 - displaying.....152
 - graceful restart.....110
 - hold-down timer.....113
 - maximum hop count.....6
 - metrics.....118, 119
 - monitoring.....301
 - neighbors.....120
 - displaying.....154
 - overview.....3, 13
 - packets.....5
 - path cost metrics *See* path cost metrics
 - poison reverse technique.....6
 - policy, routing.....109, 114
 - preferences.....122
 - reserved fields.....107
 - rib-group messages.....65
 - rib-group statement
 - usage guidelines.....65
 - route timeout.....125
 - routing policy (configuration editor).....14
 - split horizon technique.....6
 - statistics.....301
 - clearing.....137
 - displaying.....156
 - supported software standards.....9
 - tracing operations
 - authentication.....89
 - description.....89, 90
 - error PDUs.....89
 - hold-down processing.....89
 - NSR synchronization.....89
 - policy processing.....89
 - protocol task processing.....89
 - protocol timer processing.....89
 - request PDUs.....89
 - route expiration processing.....89
 - route information.....89, 90
 - state transitions.....89
 - trigger updates.....89
 - update PDUs.....89
 - traffic control with metrics *See* path cost metrics
 - metrics
 - traffic control with metrics, configuring.....53, 55
 - UDP, use of.....4
 - unidirectional limitations.....7
 - update interval.....131
 - update messages.....61, 117
 - verifying host reachability19
 - RIP (Routing Information Protocol)
 - authentication (RIPv2 only), configuring.....25
 - requirements.....8
 - verifying host reachability96
 - verifying RIP message exchange94
 - verifying RIP-enabled interfaces96
 - RIP neighbors
 - displaying.....302
 - status.....302
 - RIP neighbors, verifying.....96
 - RIP routing information.....301
 - rip statement.....124
 - RIPng
 - supported software standards.....9
 - route advertisements, displaying.....170
 - route, displaying
 - next-hop.....244
 - route-timeout statement
 - RIP.....125
 - usage guidelines.....70
 - routes, displaying
 - active.....159
 - active path.....165
 - advertising protocol.....170
 - all.....175
 - best.....177
 - brief information.....180
 - detailed information.....182
 - extensive information.....201

- hidden.....230
- in a specific routing table.....275
- in the forwarding table.....216
- inactive path.....233
- instances.....237
- learned from a specific protocol.....254
- matching the specified address.....197
- policy-based route export.....199
- received through a neighbor.....266
- sent to a specific interface.....250
- terse information.....286
- routing
 - in one AS with RIP.....13
 - neighbors See BGP peers; OSPF neighbors; RIP neighbors
 - RIP See RIP
 - RIP statistics.....94
- Routing Information Protocol See RIP
- routing policies
 - displaying.....148
 - RIP routing policy (configuration editor).....14
 - testing the configuration for.....289
- routing protocol process memory
 - FAQ.....293, 294
- routing solutions
 - controlling RIP traffic with the incoming metric.....53
 - controlling RIP traffic with the outgoing metric.....55
 - poison reverse, for traffic reduction.....6
 - split horizon, for traffic reduction.....6
- routing table
 - sample distance-vector routing.....4
 - updates, limitations in RIP.....7
- routing-instances statement.....126

S

security	
MD5 authentication for RIPv2.....	25
password authentication for RIPv2.....	25
send statement	
RIP.....	127
usage guidelines.....	61
show policy command.....	148
show policy conditions command.....	150
show rip general-statistics command.....	152
show rip neighbor command.....	96, 154
show rip neighbors command.....	301
show rip statistics command.....	94, 156, 301

show route active-path command.....	165
show route advertising-protocol command.....	170
show route all command.....	175
show route best command.....	177
show route brief command.....	180
show route command.....	159
show route detail command.....	182
show route exact command.....	197
show route export command.....	199
show route extensive command.....	201
show route forwarding-table command.....	216
show route hidden command.....	230
show route inactive-path command.....	233
show route instance command.....	237
show route next-hop command.....	244
show route output command.....	250
show route protocol command.....	254
show route receive-protocol command.....	266
show route table command.....	275
show route terse command.....	286
split horizon technique.....	6
statistics	
RIP.....	94, 301
status	
RIP neighbors.....	302
support, technical See technical support	
syntax conventions.....	xiii

T

- technical support
 - contacting JTAC.....xv
- test policy command.....289
- threshold statement
 - RIP
 - usage guidelines.....27
- topology
 - sample distance-vector routing.....4
 - sample poison reverse routing.....7
 - sample split horizon routing.....6
 - sample unidirectional routing.....7
- traceoptions statement
 - RIP.....89, 90, 128
- Traceroute page
 - results for RIP.....19, 96
- tracing flags
 - auth.....128
 - error
 - RIP.....128
 - holddown.....128

packets	
RIP.....	128
trigger.....	128
update	
RIP.....	128
tracing operations	
RIP.....	89, 90, 128
traffic	
controlling with incoming RIP metric.....	53
controlling with outgoing RIP metric.....	55
trigger (tracing flag).....	128
 U	
update (tracing flag)	
RIP.....	128
update-interval statement	
RIP.....	131
usage guidelines.....	70, 71
 V	
verification	
RIP host reachability.....	19, 96
RIP message exchange.....	94
RIP-enabled interfaces.....	96
version statement	
RIP.....	104
usage guidelines.....	27

