



---

Junos<sup>®</sup> OS

## BGP Feature Guide for Routing Devices

Release

14.1



---

Published: 2014-08-24

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

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.

*Junos<sup>®</sup> OS BGP Feature Guide for Routing Devices*

14.1

Copyright © 2014, 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 . . . . .	xxi
	Documentation and Release Notes . . . . .	xxi
	Supported Platforms . . . . .	xxi
	Using the Examples in This Manual . . . . .	xxii
	Merging a Full Example . . . . .	xxii
	Merging a Snippet . . . . .	xxiii
	Documentation Conventions . . . . .	xxiii
	Documentation Feedback . . . . .	xxv
	Requesting Technical Support . . . . .	xxvi
	Self-Help Online Tools and Resources . . . . .	xxvi
	Opening a Case with JTAC . . . . .	xxvi
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Introduction to BGP . . . . .</b>	<b>3</b>
	Understanding BGP . . . . .	4
	Autonomous Systems . . . . .	4
	AS Paths and Attributes . . . . .	4
	External and Internal BGP . . . . .	5
	Multiple Instances of BGP . . . . .	5
	BGP Routes Overview . . . . .	6
	BGP Messages Overview . . . . .	7
	Open Messages . . . . .	7
	Update Messages . . . . .	8
	Keepalive Messages . . . . .	8
	Notification Messages . . . . .	8
	Understanding BGP Path Selection . . . . .	8
	Routing Table Path Selection . . . . .	10
	Effects of Advertising Multiple Paths to a Destination . . . . .	12
	BGP Configuration Overview . . . . .	12
<b>Chapter 2</b>	<b>BGP Standards . . . . .</b>	<b>13</b>
	Supported BGP Standards . . . . .	13
<b>Chapter 3</b>	<b>Understanding 4-Byte Autonomous System Numbers . . . . .</b>	<b>17</b>
	4-Byte Autonomous System Numbers Overview . . . . .	17
	Juniper Networks Implementation of 4-Byte Autonomous System Numbers . . . . .	18
	Prepending 4-Byte AS Numbers in an AS Path . . . . .	20
	Configuring 4-Byte AS Numbers and BGP Extended Community Attributes . . . . .	21
	Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain . . . . .	22
	Understanding 4-Byte AS Numbers and Route Distinguishers . . . . .	24

	Understanding 4-Byte AS Numbers and Route Loop Detection . . . . .	25
	Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number . . . . .	27
	Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number . . . . .	28
<b>Part 2</b>	<b>Configuration</b>	
<b>Chapter 4</b>	<b>Basic BGP Configuration . . . . .</b>	<b>33</b>
	Examples: Configuring External BGP Peering . . . . .	33
	Understanding External BGP Peering Sessions . . . . .	33
	Example: Configuring External BGP Point-to-Point Peer Sessions . . . . .	34
	Example: Configuring External BGP on Logical Systems with IPv6 Interfaces . . . . .	41
	Examples: Configuring Internal BGP Peering . . . . .	56
	Understanding Internal BGP Peering Sessions . . . . .	56
	Example: Configuring Internal BGP Peer Sessions . . . . .	57
	Example: Configuring Internal BGP Peering Sessions on Logical Systems . .	68
<b>Chapter 5</b>	<b>BGP Path Attribute Configuration . . . . .</b>	<b>79</b>
	Example: Configuring BGP Local Preference . . . . .	79
	Understanding the BGP Local Preference . . . . .	79
	Example: Configuring the Local Preference Value for BGP Routes . . . . .	80
	Examples: Configuring BGP MED . . . . .	92
	Understanding the MED Attribute . . . . .	92
	Example: Configuring the MED Attribute Directly . . . . .	95
	Example: Configuring the MED Using Route Filters . . . . .	107
	Example: Configuring the MED Using Communities . . . . .	120
	Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates . . . . .	121
	Examples: Configuring BGP Local AS . . . . .	131
	Understanding the BGP Local AS Attribute . . . . .	131
	Example: Configuring a Local AS for EBGp Sessions . . . . .	136
	Example: Configuring a Private Local AS for EBGp Sessions . . . . .	146
	Example: Configuring the Accumulated IGP Attribute for BGP . . . . .	151
	Understanding the Accumulated IGP Attribute for BGP . . . . .	152
	Example: Configuring the Accumulated IGP Attribute for BGP . . . . .	152
	Example: Configuring AS Override . . . . .	190
	Understanding AS Override . . . . .	190
	Example: Configuring a Layer 3 VPN with Route Reflection and AS Override . . . . .	191
	Example: Disabling Suppression of Route Advertisements . . . . .	200
	Configuring 4-Byte Autonomous System Numbers . . . . .	208
	Disabling Attribute Set Messages on Independent AS Domains for BGP Loop Detection . . . . .	209

<b>Chapter 6</b>	<b>BGP Policy Configuration . . . . .</b>	<b>211</b>
	Example: Applying Routing Policies at Different Levels of the BGP Hierarchy . . .	211
	Example: Configuring BGP Interactions with IGPs . . . . .	220
	Understanding Routing Policies . . . . .	220
	Example: Injecting OSPF Routes into the BGP Routing Table . . . . .	220
	Example: Redistributing BGP Routes with a Specific Community Tag into	
	IS-IS . . . . .	223
	Understanding BGP Communities and Extended Communities as Routing	
	Policy Match Conditions . . . . .	223
	Example: Redistributing BGP Routes with a Specific Community Tag into	
	IS-IS . . . . .	225
	Example: Configuring BGP Route Advertisement . . . . .	233
	Understanding Route Advertisement . . . . .	233
	Applying Routing Policy . . . . .	233
	Setting BGP to Advertise Inactive Routes . . . . .	234
	Configuring BGP to Advertise the Best External Route to Internal	
	Peers . . . . .	235
	Configuring How Often BGP Exchanges Routes with the Routing	
	Table . . . . .	236
	Disabling Suppression of Route Advertisements . . . . .	237
	Example: Configuring BGP Prefix-Based Outbound Route Filtering . . . . .	237
	Example: Configuring EBGp Multihop . . . . .	241
	Understanding BGP Multihop . . . . .	241
	Example: Configuring EBGp Multihop Sessions . . . . .	241
	Example: Configuring BGP Route Preference (Administrative Distance) . . . . .	250
	Understanding Route Preference Values . . . . .	250
	Example: Configuring the Preference Value for BGP Routes . . . . .	252
	Example: Configuring BGP Path Selection . . . . .	257
	Understanding BGP Path Selection . . . . .	257
	Routing Table Path Selection . . . . .	259
	Effects of Advertising Multiple Paths to a Destination . . . . .	260
	Example: Ignoring the AS Path Attribute When Selecting the Best Path . . .	260
	Example: Removing Private AS Numbers . . . . .	267
	Understanding Private AS Number Removal from AS Paths . . . . .	268
	Example: Removing Private AS Numbers from AS Paths . . . . .	269
	Example: Overriding the Default BGP Routing Policy on PTX Series Packet	
	Transport Routers . . . . .	274
	Understanding the Default BGP Routing Policy on Packet Transport	
	Routers . . . . .	274
	Example: Overriding the Default BGP Routing Policy on PTX Series Packet	
	Transport Routers . . . . .	276
	Example: Configuring Conditional Installation of Prefixes in a Routing Table . . .	278
	Conditional Installation of Prefixes Use Cases . . . . .	279
	Understanding Conditional Installation of Prefixes in a Routing Table . . . .	281
	Example: Configuring Conditional Installation of Prefixes in a Routing	
	Table . . . . .	282
	Example: Setting BGP to Advertise Inactive Routes . . . . .	298
	Example: Configuring BGP to Advertise the Best External Route to Internal	
	Peers . . . . .	304

	Example: Disabling Suppression of Route Advertisements . . . . .	312
	Example: Defining a Routing Policy That Removes BGP Communities . . . . .	319
	Example: Defining a Routing Policy Based on the Number of BGP Communities . . . . .	326
	Example: Using Routing Policy to Set a Preference Value for BGP Routes . . . . .	333
	Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths . . . . .	339
<b>Chapter 7</b>	<b>BGP BFD Configuration . . . . .</b>	<b>349</b>
	Example: Configuring BFD for BGP . . . . .	349
	Understanding BFD for BGP . . . . .	349
	Example: Configuring BFD on Internal BGP Peer Sessions . . . . .	350
	Example: Configuring BFD Authentication for BGP . . . . .	358
	Understanding BFD Authentication for BGP . . . . .	358
	BFD Authentication Algorithms . . . . .	359
	Security Authentication Keychains . . . . .	360
	Strict Versus Loose Authentication . . . . .	360
	Example: Configuring BFD Authentication for BGP . . . . .	360
	Configuring BFD Authentication Parameters . . . . .	360
	Viewing Authentication Information for BFD Sessions . . . . .	362
<b>Chapter 8</b>	<b>BGP Load Balancing Configuration . . . . .</b>	<b>365</b>
	Examples: Configuring BGP Multipath . . . . .	365
	Understanding BGP Multipath . . . . .	365
	Example: Load Balancing BGP Traffic . . . . .	366
	Example: Configuring Single-Hop EBGP Peers to Accept Remote Next Hops . . . . .	371
	Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths . . . . .	382
	Understanding Load Balancing for BGP Traffic with Unequal Bandwidth Allocated to the Paths . . . . .	382
	Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths . . . . .	383
	Example: Advertising Multiple BGP Paths to a Destination . . . . .	392
	Understanding the Advertisement of Multiple Paths to a Single Destination in BGP . . . . .	392
	Example: Advertising Multiple Paths in BGP . . . . .	393
<b>Chapter 9</b>	<b>IBGP Scaling Configuration . . . . .</b>	<b>419</b>
	Example: Configuring BGP Route Reflectors . . . . .	419
	Understanding BGP Route Reflectors . . . . .	419
	Example: Configuring a Route Reflector . . . . .	421
	Example: Configuring a Route Reflector that Belongs to Two Different Clusters . . . . .	436
	Understanding a Route Reflector that Belongs to Two Different Clusters . . . . .	436
	Example: Configuring a Route Reflector that Belongs to Two Different Clusters . . . . .	437
	Example: Configuring BGP Confederations . . . . .	441
	Understanding BGP Confederations . . . . .	441
	Example: Configuring BGP Confederations . . . . .	442

<b>Chapter 10</b>	<b>BGP Security Configuration . . . . .</b>	<b>449</b>
	Example: Configuring BGP Route Authentication . . . . .	449
	Understanding Route Authentication . . . . .	449
	Example: Configuring Route Authentication for BGP . . . . .	450
	Example: Configuring IPsec Protection for BGP . . . . .	456
	Understanding IPsec for BGP . . . . .	456
	Example: Using IPsec to Protect BGP Traffic . . . . .	456
	Examples: Configuring TCP and BGP Security . . . . .	459
	Understanding Security Options for BGP with TCP . . . . .	460
	Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers . . . . .	460
	Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List . . . . .	465
	Example: Limiting TCP Segment Size for BGP . . . . .	468
	Example: Configuring Origin Validation for BGP . . . . .	473
	Use Case and Benefit of Origin Validation . . . . .	473
	Understanding Origin Validation for BGP . . . . .	474
	Supported Standards . . . . .	475
	How Origin Validation Works . . . . .	475
	BGP Interaction with the Route Validation Database . . . . .	477
	Community Attribute to Announce RPKI Validation State to IBGP Neighbors . . . . .	479
	Nonstop Active Routing and Origin Validation . . . . .	480
	Marking a Prefix Range as Never Allowed . . . . .	480
	Example: Configuring Origin Validation for BGP . . . . .	480
<b>Chapter 11</b>	<b>BGP Flap Configuration . . . . .</b>	<b>497</b>
	Example: Preventing BGP Session Resets . . . . .	497
	Understanding BGP Session Resets . . . . .	497
	Example: Preventing BGP Session Flaps When VPN Families Are Configured . . . . .	497
	Examples: Configuring BGP Flap Damping . . . . .	505
	Understanding BGP Route Flap Damping Parameters . . . . .	505
	Example: Configuring BGP Route Flap Damping Parameters . . . . .	506
	Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family . . . . .	515
	Example: Configuring Error Handling for BGP Update Messages . . . . .	525
	Understanding Error Handling for BGP Update Messages . . . . .	525
	Example: Configuring Error Handling for BGP Update Messages . . . . .	527
<b>Chapter 12</b>	<b>Multiprotocol BGP Configuration . . . . .</b>	<b>537</b>
	Examples: Configuring Multiprotocol BGP . . . . .	537
	Understanding Multiprotocol BGP . . . . .	537
	Limiting the Number of Prefixes Received on a BGP Peer Session . . . . .	541
	Limiting the Number of Prefixes Accepted on a BGP Peer Session . . . . .	541
	Configuring BGP Routing Table Groups . . . . .	542
	Resolving Routes to PE Routing Devices Located in Other ASs . . . . .	542

	Allowing Labeled and Unlabeled Routes . . . . .	543
	Example: Configuring IPv6 BGP Routes over IPv4 Transport . . . . .	543
	Enabling Layer 2 VPN and VPLS Signaling . . . . .	549
	Example: Configuring Flow Routes . . . . .	550
	Understanding Flow Routes . . . . .	550
	Match Conditions for Flow Routes . . . . .	550
	Actions for Flow Routes . . . . .	552
	Validating Flow Routes . . . . .	553
	Support for BGP Flow-Specification Algorithm Version 7 and Later . . . . .	553
	Example: Enabling BGP to Carry Flow-Specification Routes . . . . .	554
<b>Chapter 13</b>	<b>BGP CLNS Configuration . . . . .</b>	<b>569</b>
	Example: Configuring BGP and CLNS . . . . .	569
	Understanding BGP for CLNS VPNs . . . . .	569
	Example: Configuring BGP for CLNS VPNs . . . . .	569
	Enabling BGP to Carry CLNS Routes . . . . .	571
	Example: Enabling CLNS Between Two Routers . . . . .	572
	Example: Configuring CLNS Within a VPN . . . . .	574
<b>Chapter 14</b>	<b>BGP Monitoring Configuration . . . . .</b>	<b>577</b>
	Configuring BGP Monitoring Protocol Version 3 . . . . .	577
	Example: Configuring BGP Monitoring Protocol . . . . .	580
	Understanding the BGP Monitoring Protocol . . . . .	580
	Example: Configuring the BGP Monitoring Protocol . . . . .	580
	Example: Configuring BGP Trace Operations . . . . .	583
	Understanding Trace Operations for BGP Protocol Traffic . . . . .	583
	Example: Viewing BGP Trace Files on Logical Systems . . . . .	584
	Tracing BMP Operations . . . . .	589
<b>Chapter 15</b>	<b>BGP Configuration Statements . . . . .</b>	<b>593</b>
	[edit protocols bgp] Hierarchy Level . . . . .	597
	Common BGP Family Options . . . . .	597
	Complete [edit protocols bgp] Hierarchy . . . . .	597
	accept-remote-nexthop . . . . .	604
	accepted-prefix-limit . . . . .	605
	add-path . . . . .	607
	advertise-external . . . . .	608
	advertise-from-main-vpn-tables . . . . .	610
	advertise-inactive . . . . .	611
	advertise-peer-as . . . . .	612
	aggregate-label . . . . .	613
	aigp . . . . .	614
	aigp-originate . . . . .	616
	algorithm (BGP BFD Authentication) . . . . .	617
	allow . . . . .	619
	as-override . . . . .	620
	authentication (BGP BFD Liveness Detection) . . . . .	621
	authentication-algorithm . . . . .	623
	authentication-key (Protocols BGP and BMP) . . . . .	624
	authentication-key-chain (Protocols BGP and BMP) . . . . .	625

auto-discovery-only	626
bfd-liveness-detection (Protocols BGP)	627
bgp	631
bgp-error-tolerance (Protocols BGP)	631
bgp-orf-cisco-mode	632
bmp	634
cluster	636
connection-mode	637
damping (Protocols BGP)	638
description (Protocols BGP)	640
detection-time (BFD Liveness Detection)	641
disable (Protocols BGP)	642
disable (BGP Graceful Restart)	643
explicit-null (Protocols BGP)	644
export (Protocols BGP)	646
family (Protocols BGP)	647
file (Tracing for Origin AS Validation)	651
flag (Tracing for Origin AS Validation)	652
flow	653
graceful-restart (Protocols BGP)	654
group (Protocols BGP)	655
group (Origin Validation for BGP)	658
hold-down	659
hold-down-interval (BGP BFD Liveness Detection)	661
hold-time (Protocols BGP)	663
hold-time (Origin Validation for BGP)	665
idle-after-switch-over	666
import (Protocols BGP)	667
include-mp-next-hop	669
inet-mdt (Signaling)	670
initiation-message	671
ipsec-sa (Protocols BGP)	672
iso-vpn	673
keep	674
key-chain (BGP BFD Authentication)	676
labeled-unicast (Protocols BGP)	678
local-address (Protocols BGP)	680
local-address (Protocols BMP)	682
local-address (Origin Validation for BGP)	683
local-as	684
local-interface (IPv6)	686
local-port	687
local-preference	688
log-updown (Protocols BGP)	689
logical-systems	690
loops	691
loose-check (BGP BFD Authentication)	693
malformed-route-limit (Protocols BGP)	694
malformed-update-log-interval (Protocols BGP)	695

max-sessions (Origin Validation for BGP) .....	696
maximum-length (Origin Validation for BGP) .....	697
metric-out (Protocols BGP) .....	698
minimum-interval (BFD Liveness Detection) .....	700
minimum-interval (transmit-interval) .....	702
minimum-receive-interval (BFD Liveness Detection) .....	704
monitor (Protocols BMP) .....	705
mtu-discovery .....	706
multihop .....	708
multipath (Protocols BGP) .....	710
multiplier (BFD Liveness Detection) .....	711
neighbor (Protocols BGP) .....	713
no-adaptation (BFD Liveness Detection) .....	716
no advertise-peer-as .....	717
no-aggregator-id .....	718
no-client-reflect .....	719
no-malformed-route-limit (Protocols BGP) .....	720
no-nexthop-change (BGP multihop) .....	721
no-validate .....	723
origin-autonomous-system (Origin Validation for BGP) .....	724
out-delay .....	725
outbound-route-filter .....	727
passive (Protocols BGP) .....	728
path-count .....	729
path-selection .....	730
peer-as (Protocols BGP) .....	732
port (Origin Validation for BGP) .....	733
pre-policy .....	734
precision-timers .....	735
preference (Protocols BGP) .....	736
preference (Origin Validation for BGP) .....	737
prefix-limit .....	738
prefix-policy .....	740
priority (Protocols BMP) .....	741
post-policy .....	742
protection (Protocols BGP) .....	742
protection (Protocols MPLS) .....	743
protocols .....	744
receive (Protocols BGP) .....	746
record (Origin Validation for BGP) .....	747
record-lifetime (Origin Validation for BGP) .....	748
refresh-time (Origin Validation for BGP) .....	749
remove-private .....	750
resolve-vpn .....	752
restart-time (BGP Graceful Restart) .....	753
rib (Protocols BGP) .....	754
rib-group (Protocols BGP) .....	755
route-monitoring .....	756
route-target (Protocols BGP) .....	757

routing-instances (Multiple Routing Entities) . . . . .	758
send (Logical Systems Add-Path) . . . . .	759
session (Origin Validation for BGP) . . . . .	760
session-mode . . . . .	761
stale-routes-time . . . . .	762
static (Origin Validation for BGP) . . . . .	763
station . . . . .	764
station-address . . . . .	765
station-port . . . . .	766
statistics-timeout . . . . .	767
tcp-aggressive-transmission . . . . .	768
tcp-mss (Protocols BGP) . . . . .	769
threshold (detection-time) . . . . .	770
threshold (transmit-interval) . . . . .	772
topology (Protocols BGP) . . . . .	774
traceoptions (Protocols BGP) . . . . .	776
traceoptions (Protocols BMP) . . . . .	779
traceoptions (Origin Validation for BGP) . . . . .	781
traffic-statistics (Protocols BGP) . . . . .	782
transmit-interval (BFD Liveness Detection) . . . . .	783
ttl (Protocols BGP) . . . . .	785
type (Protocols BGP) . . . . .	787
validation (Origin Validation for BGP) . . . . .	788
validation-state (Origin Validation for BGP) . . . . .	789
version (BFD Liveness Detection) . . . . .	790
vpn-apply-export . . . . .	791

## Part 3

### Chapter 16

## Administration

<b>BGP Operational Commands . . . . .</b>	<b>795</b>
clear bfd adaptation . . . . .	797
clear bfd session . . . . .	798
clear bgp damping . . . . .	799
clear bgp neighbor . . . . .	800
clear bgp table . . . . .	802
clear validation database . . . . .	804
clear validation session . . . . .	805
clear validation statistics . . . . .	806
monitor traffic . . . . .	807
request validation policy . . . . .	817
restart . . . . .	818
show bfd session . . . . .	829
show bgp bmp . . . . .	837
show bgp group . . . . .	839
show bgp group traffic-statistics . . . . .	846
show bgp neighbor . . . . .	848
show bgp replication . . . . .	862
show bgp replication logical-system . . . . .	865
show bgp summary . . . . .	867

show policy damping . . . . .	873
show policy . . . . .	875
show policy conditions . . . . .	877
show policy damping . . . . .	879
show route . . . . .	881
show route active-path . . . . .	887
show route advertising-protocol . . . . .	892
show route all . . . . .	897
show route aspath-regex . . . . .	899
show route best . . . . .	901
show route brief . . . . .	904
show route community . . . . .	906
show route community-name . . . . .	908
show route damping . . . . .	910
show route detail . . . . .	915
show route exact . . . . .	932
show route export . . . . .	934
show route extensive . . . . .	937
show route flow validation . . . . .	954
show route forwarding-table . . . . .	956
show route hidden . . . . .	970
show route inactive-path . . . . .	973
show route inactive-prefix . . . . .	976
show route instance . . . . .	978
show route next-hop . . . . .	986
show route no-community . . . . .	992
show route output . . . . .	995
show route protocol . . . . .	1000
show route receive-protocol . . . . .	1012
show route table . . . . .	1020
show route terse . . . . .	1034
show security keychain . . . . .	1037
show validation database . . . . .	1040
show validation group . . . . .	1042
show validation replication database . . . . .	1044
show validation session . . . . .	1046
show validation statistics . . . . .	1049
test policy . . . . .	1051

## Part 4

### Chapter 17

## Troubleshooting

<b>BGP Troubleshooting . . . . .</b>	<b>1055</b>
Understanding Hidden Routes . . . . .	1056
Checklist for Verifying the BGP Protocol and Peers . . . . .	1057
Checklist for Checking the BGP Layer . . . . .	1058
Checking the BGP Layer . . . . .	1058
Check That BGP Traffic Is Using the LSP . . . . .	1060
Check BGP Sessions . . . . .	1061
Verify the BGP Configuration . . . . .	1062

	Examine BGP Routes . . . . .	1068
	Verify Received BGP Routes . . . . .	1069
	Take Appropriate Action . . . . .	1070
	Check That BGP Traffic Is Using the LSP Again . . . . .	1071
	Check BGP Sessions . . . . .	1072
	Verify BGP Peers . . . . .	1074
	Verify the BGP Protocol . . . . .	1075
	Verify the BGP Configuration . . . . .	1077
	Display Sent or Received BGP Packets . . . . .	1082
	Diagnose BGP Session Establishment Problems . . . . .	1083
	Examine BGP Routes and Route Selection . . . . .	1084
	Examine the Local Preference Selection . . . . .	1086
	Examine the Multiple Exit Discriminator Route Selection . . . . .	1087
	Examine the EBGp over IBGP Selection . . . . .	1088
	Examine the IGP Cost Selection . . . . .	1089
	Examine Routes in the Forwarding Table . . . . .	1090
	Log BGP State Transition Events . . . . .	1091
	Display Detailed BGP Protocol Information . . . . .	1093
	Verify Received BGP Routes . . . . .	1095
	Verify That a Particular BGP Route Is Received on Your Router . . . . .	1096
	Verifying Advertised BGP Routes . . . . .	1096
	Check That BGP Traffic Is Using the LSP . . . . .	1097
	Check That BGP Traffic Is Using the LSP Again . . . . .	1097
	Examine the EBGp over IBGP Selection . . . . .	1098
	Verify BGP on an Internal Router . . . . .	1099
	Verify BGP on a Border Router . . . . .	1102
<b>Chapter 18</b>	<b>Routing Protocol Process Memory FAQs . . . . .</b>	<b>1107</b>
	Routing Protocol Process Memory FAQs Overview . . . . .	1107
	Routing Protocol Process Memory FAQs . . . . .	1108
	Frequently Asked Questions: Routing Protocol Process Memory . . . . .	1108
	Frequently Asked Questions: Interpreting Routing Protocol Process-Related Command Outputs . . . . .	1109
	Frequently Asked Questions: Routing Protocol Process Memory Swapping . . . . .	1112
	Frequently Asked Questions: Troubleshooting the Routing Protocol Process . . . . .	1113
<b>Part 5</b>	<b>Index</b>	
	Index . . . . .	1117



# List of Figures

<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Introduction to BGP</b>	<b>3</b>
	Figure 1: ASs, EBGp, and IBGP	5
<b>Chapter 3</b>	<b>Understanding 4-Byte Autonomous System Numbers</b>	<b>17</b>
	Figure 2: EBGp With 4-Byte AS Numbers Prepended to the AS Path	20
	Figure 3: 4-Byte Capable Router AS Path to a 2-Byte Capable Router	22
	Figure 4: EBGp 4-Byte AS Path Through a 2-Byte AS Domain	22
	Figure 5: IBGP 4-Byte AS Path Through a 2-Byte AS Domain	23
	Figure 6: 4-Byte AS Numbers and Loop Detection	26
	Figure 7: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 2-Byte AS Number	27
	Figure 8: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 4-Byte AS Number	29
<b>Part 2</b>	<b>Configuration</b>	
<b>Chapter 4</b>	<b>Basic BGP Configuration</b>	<b>33</b>
	Figure 9: BGP Peering Session	33
	Figure 10: Typical Network with BGP Peer Sessions	35
	Figure 11: Typical Network with BGP Peer Sessions	42
	Figure 12: Internal and External BGP	56
	Figure 13: Typical Network with IBGP Sessions	59
	Figure 14: Typical Network with IBGP Sessions	69
<b>Chapter 5</b>	<b>BGP Path Attribute Configuration</b>	<b>79</b>
	Figure 15: Typical Network with IBGP Sessions and Multiple Exit Points	80
	Figure 16: Default MED Example	93
	Figure 17: Typical Network with IBGP Sessions and Multiple Exit Points	96
	Figure 18: Typical Network with IBGP Sessions and Multiple Exit Points	108
	Figure 19: Topology for Delaying the MED Update	123
	Figure 20: Local AS Configuration	134
	Figure 21: Topology for Configuring the Local AS	137
	Figure 22: Topology for Configuring a Private Local AS	147
	Figure 23: Advertisement of Multiple Paths in BGP	154
	Figure 24: AS Override Topology	191
	Figure 25: BGP Topology for advertise-peer-as	202
<b>Chapter 6</b>	<b>BGP Policy Configuration</b>	<b>211</b>
	Figure 26: Applying Routing Policies to BGP	213

	Figure 27: Redistributing BGP Routes with a Specific Community Tag into IS-IS . . . . .	225
	Figure 28: BGP Prefix-Based Outbound Route Filtering . . . . .	238
	Figure 29: Typical Network with EBGP Multihop Sessions . . . . .	242
	Figure 30: BGP Preference Value Topology . . . . .	253
	Figure 31: Topology for Ignoring the AS-Path Length . . . . .	262
	Figure 32: Topology for Removing a Private AS from the Advertised AS Path . . . . .	269
	Figure 33: BGP Import and Export Policies . . . . .	281
	Figure 34: Conditional Installation of Prefixes . . . . .	285
	Figure 35: BGP Topology for advertise-inactive . . . . .	299
	Figure 36: BGP Topology for advertise-external . . . . .	306
	Figure 37: BGP Topology for advertise-peer-as . . . . .	313
	Figure 38: BGP Policy That Removes Communities . . . . .	320
	Figure 39: BGP Policy with a Limit on the Number of Communities Accepted . . . . .	327
	Figure 40: BGP Preference Value Topology . . . . .	335
	Figure 41: BGP Load Balancing . . . . .	341
<b>Chapter 7</b>	<b>BGP BFD Configuration . . . . .</b>	<b>349</b>
	Figure 42: Typical Network with IBGP Sessions . . . . .	351
<b>Chapter 8</b>	<b>BGP Load Balancing Configuration . . . . .</b>	<b>365</b>
	Figure 43: BGP Load Balancing . . . . .	367
	Figure 44: Topology for Accepting a Remote Next Hop . . . . .	372
	Figure 45: BGP Load Balancing . . . . .	385
	Figure 46: Advertisement of Multiple Paths in BGP . . . . .	394
<b>Chapter 9</b>	<b>IBGP Scaling Configuration . . . . .</b>	<b>419</b>
	Figure 47: Simple Route Reflector Topology (One Cluster) . . . . .	420
	Figure 48: Basic Route Reflection (Multiple Clusters) . . . . .	420
	Figure 49: Hierarchical Route Reflection (Clusters of Clusters) . . . . .	421
	Figure 50: IBGP Network Using a Route Reflector . . . . .	423
	Figure 51: Route Reflector in Two Different Clusters . . . . .	438
	Figure 52: BGP Confederations . . . . .	442
	Figure 53: Typical Network Using BGP Confederations . . . . .	443
<b>Chapter 10</b>	<b>BGP Security Configuration . . . . .</b>	<b>449</b>
	Figure 54: Authentication for BGP . . . . .	452
	Figure 55: IPsec for BGP . . . . .	457
	Figure 56: Typical Network with BGP Peer Sessions . . . . .	461
	Figure 57: TCP Maximum Segment Size for BGP . . . . .	469
	Figure 58: Sample Topology for Origin Validation . . . . .	475
	Figure 59: BGP and Route Validation . . . . .	478
	Figure 60: Importing and Exporting Routing Policies . . . . .	478
	Figure 61: Topology for Origin Validation . . . . .	482
<b>Chapter 11</b>	<b>BGP Flap Configuration . . . . .</b>	<b>497</b>
	Figure 62: Topology for the EBGP Case . . . . .	501
	Figure 63: Topology for the RR Case . . . . .	501
	Figure 64: BGP Flap Damping Topology . . . . .	507
	Figure 65: MBGP MVPN with BGP Route Flap Damping . . . . .	515
	Figure 66: BGP Error Handling Example Topology . . . . .	528

<b>Chapter 12</b>	<b>Multiprotocol BGP Configuration . . . . .</b>	<b>537</b>
	Figure 67: Topology for Configuring IPv6 BGP Routes over IPv4 Transport . . . . .	544
<b>Chapter 14</b>	<b>BGP Monitoring Configuration . . . . .</b>	<b>577</b>
	Figure 68: BMP Topology . . . . .	581
<b>Part 4</b>	<b>Troubleshooting</b>	
<b>Chapter 17</b>	<b>BGP Troubleshooting . . . . .</b>	<b>1055</b>
	Figure 69: Checking the BGP Layer . . . . .	1059
	Figure 70: MPLS Network Broken at the BGP Layer . . . . .	1060
	Figure 71: BGP Network Topology . . . . .	1074
	Figure 72: BGP Configuration Topology . . . . .	1076
	Figure 73: BGP Network Topology . . . . .	1085



# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xxi</b>
	Table 1: Notice Icons . . . . .	xxiv
	Table 2: Text and Syntax Conventions . . . . .	xxiv
<b>Part 2</b>	<b>Configuration</b>	
<b>Chapter 5</b>	<b>BGP Path Attribute Configuration</b> . . . . .	<b>79</b>
	Table 3: MED Options for Routing Table Path Selection . . . . .	94
<b>Chapter 6</b>	<b>BGP Policy Configuration</b> . . . . .	<b>211</b>
	Table 4: Default Route Preference Values . . . . .	250
<b>Chapter 9</b>	<b>IBGP Scaling Configuration</b> . . . . .	<b>419</b>
	Table 5: Rules for Route Reflectors . . . . .	437
<b>Chapter 11</b>	<b>BGP Flap Configuration</b> . . . . .	<b>497</b>
	Table 6: Damping Parameters . . . . .	506
<b>Chapter 12</b>	<b>Multiprotocol BGP Configuration</b> . . . . .	<b>537</b>
	Table 7: Flow Route Match Conditions . . . . .	551
	Table 8: Flow Route Action Modifiers . . . . .	552
<b>Part 3</b>	<b>Administration</b>	
<b>Chapter 16</b>	<b>BGP Operational Commands</b> . . . . .	<b>795</b>
	Table 9: Match Conditions for the monitor traffic Command . . . . .	809
	Table 10: Logical Operators for the monitor traffic Command . . . . .	810
	Table 11: Arithmetic and Relational Operators for the monitor traffic Command . . . . .	812
	Table 12: show bfd session Output Fields . . . . .	830
	Table 13: show bgp bmp Output Fields . . . . .	837
	Table 14: show bgp group Output Fields . . . . .	840
	Table 15: show bgp group traffic-statistics Output Fields . . . . .	846
	Table 16: show bgp neighbor Output Fields . . . . .	849
	Table 17: show bgp replication Output Fields . . . . .	862
	Table 18: show bgp replication logical-system Output Fields . . . . .	865
	Table 19: show bgp summary Output Fields . . . . .	868
	Table 20: show policy damping Output Fields . . . . .	874
	Table 21: show policy Output Fields . . . . .	875
	Table 22: show policy conditions Output Fields . . . . .	877
	Table 23: show policy damping Output Fields . . . . .	880
	Table 24: show route Output Fields . . . . .	882

	Table 25: show route advertising-protocol Output Fields . . . . .	893
	Table 26: show route damping Output Fields . . . . .	911
	Table 27: show route detail Output Fields . . . . .	915
	Table 28: Next-hop Types Output Field Values . . . . .	920
	Table 29: State Output Field Values . . . . .	921
	Table 30: Communities Output Field Values . . . . .	923
	Table 31: show route export Output Fields . . . . .	934
	Table 32: show route extensive Output Fields . . . . .	937
	Table 33: show route flow validation Output Fields . . . . .	954
	Table 34: show route forwarding-table Output Fields . . . . .	959
	Table 35: show route instance Output Fields . . . . .	979
	Table 36: show route receive-protocol Output Fields . . . . .	1012
	Table 37: show route terse Output Fields . . . . .	1034
	Table 38: show security keychain Output Fields . . . . .	1037
	Table 39: show validation database Output Fields . . . . .	1041
	Table 40: show validation group Output Fields . . . . .	1042
	Table 41: show validation replication database Output Fields . . . . .	1045
	Table 42: show validation session Output Fields . . . . .	1046
	Table 43: show validation statistics Output Fields . . . . .	1049
<b>Part 4</b>	<b>Troubleshooting</b>	
<b>Chapter 17</b>	<b>BGP Troubleshooting . . . . .</b>	<b>1055</b>
	Table 44: Checklist for Verifying the BGP Protocol and Peers . . . . .	1057
	Table 45: Checklist for Checking the BGP Layer . . . . .	1058
	Table 46: Six States of a BGP Session . . . . .	1092
	Table 47: BGP Protocol Tracing Flags . . . . .	1094
<b>Chapter 18</b>	<b>Routing Protocol Process Memory FAQs . . . . .</b>	<b>1107</b>
	Table 48: show system processes extensive Output Fields . . . . .	1110
	Table 49: show task memory Output Fields . . . . .	1111

# About the Documentation

- Documentation and Release Notes on page xxi
- Supported Platforms on page xxi
- Using the Examples in This Manual on page xxii
- Documentation Conventions on page xxiii
- Documentation Feedback on page xxv
- Requesting Technical Support on page xxvi

## 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
- J Series
- SRX Series
- T Series
- MX Series
- M 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 xxiv](#) 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.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies guide names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS CLI User Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<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@# <b>set system domain-name</b> <i>domain-name</i>

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

Convention	Description	Examples
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"><li>To configure a stub area, include the <b>stub</b> statement at the <b>[edit protocols ospf area area-id]</b> hierarchy level.</li><li>The console port is labeled <b>CONSOLE</b>.</li></ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub &lt;default-metric <i>metric</i>&gt;;</b>
(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.	<b>broadcast   multicast</b>  <b>(<i>string1</i>   <i>string2</i>   <i>string3</i>)</b>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members [ <i>community-ids</i> ]</b>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	<pre>[edit] routing-options {   static {     route default {       nexthop <i>address</i>;       retain;     }   } }</pre>
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"><li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li><li>To cancel the configuration, click <b>Cancel</b>.</li></ul>
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>.

- E-mail—Send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net). Include the document or topic name, URL or page number, and software 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: <http://kb.juniper.net/InfoCenter/>
- 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 BGP on page 3](#)
- [BGP Standards on page 13](#)
- [Understanding 4-Byte Autonomous System Numbers on page 17](#)



## CHAPTER 1

# Introduction to BGP

- [Understanding BGP on page 4](#)
- [BGP Routes Overview on page 6](#)
- [BGP Messages Overview on page 7](#)
- [Understanding BGP Path Selection on page 8](#)
- [BGP Configuration Overview on page 12](#)

## Understanding BGP

---

BGP is an exterior gateway protocol (EGP) that is used to exchange routing information among routers in different autonomous systems (ASs). BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network reachability information, which it exchanges with other BGP systems. BGP uses the network reachability information to construct a graph of AS connectivity, which enables BGP to remove routing loops and enforce policy decisions at the AS level.

Multiprotocol BGP (MBGP) extensions enable BGP to support IP version 6 (IPv6). MBGP defines the attributes MP\_REACH\_NLRI and MP\_UNREACH\_NLRI, which are used to carry IPv6 reachability information. Network layer reachability information (NLRI) update messages carry IPv6 address prefixes of feasible routes.

BGP allows for policy-based routing. You can use routing policies to choose among multiple paths to a destination and to control the redistribution of routing information.

BGP uses TCP as its transport protocol, using port 179 for establishing connections. Running over a reliable transport protocol eliminates the need for BGP to implement update fragmentation, retransmission, acknowledgment, and sequencing.

The Junos OS routing protocol software supports BGP version 4. This version of BGP adds support for Classless Interdomain Routing (CIDR), which eliminates the concept of network classes. Instead of assuming which bits of an address represent the network by looking at the first octet, CIDR allows you to explicitly specify the number of bits in the network address, thus providing a means to decrease the size of the routing tables. BGP version 4 also supports aggregation of routes, including the aggregation of AS paths.

This section discusses the following topics:

- [Autonomous Systems on page 4](#)
- [AS Paths and Attributes on page 4](#)
- [External and Internal BGP on page 5](#)
- [Multiple Instances of BGP on page 5](#)

### Autonomous Systems

An *autonomous system* (AS) is a set of routers that are under a single technical administration and normally use a single interior gateway protocol and a common set of metrics to propagate routing information within the set of routers. To other ASs, an AS appears to have a single, coherent interior routing plan and presents a consistent picture of what destinations are reachable through it.

### AS Paths and Attributes

The routing information that BGP systems exchange includes the complete route to each destination, as well as additional information about the route. The route to each destination is called the *AS path*, and the additional route information is included in *path attributes*. BGP uses the AS path and the path attributes to completely determine the network topology. Once BGP understands the topology, it can detect and eliminate

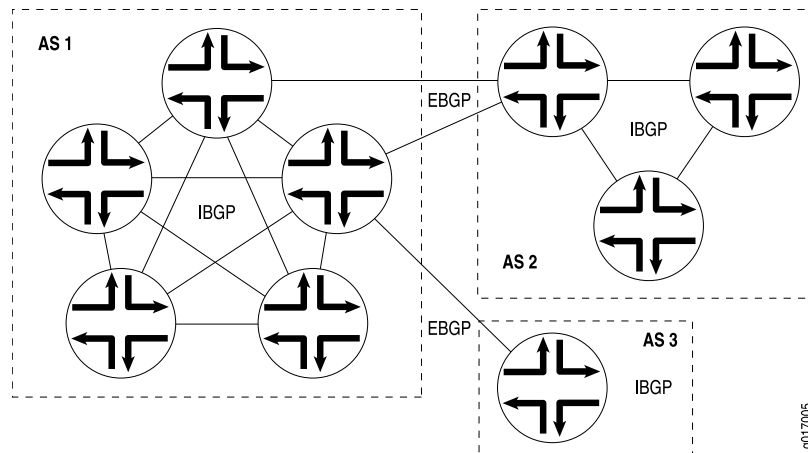
routing loops and select among groups of routes to enforce administrative preferences and routing policy decisions.

## External and Internal BGP

BGP supports two types of exchanges of routing information: exchanges among different ASs and exchanges within a single AS. When used among ASs, BGP is called *external BGP* (EBGP) and BGP sessions perform *inter-AS routing*. When used within an AS, BGP is called *internal BGP* (IBGP) and BGP sessions perform *intra-AS routing*.

Figure 1 on page 5 illustrates ASs, IBGP, and EBGP.

Figure 1: ASs, EBGP, and IBGP



A BGP system shares network reachability information with adjacent BGP systems, which are referred to as *neighbors* or *peers*.

BGP systems are arranged into *groups*. In an IBGP group, all peers in the group—called *internal peers*—are in the same AS. Internal peers can be anywhere in the local AS and do not have to be directly connected to one another. Internal groups use routes from an IGP to resolve forwarding addresses. They also propagate external routes among all other internal routers running IBGP, computing the next hop by taking the BGP next hop received with the route and resolving it using information from one of the interior gateway protocols.

In an EBGP group, the peers in the group—called *external peers*—are in different ASs and normally share a subnet. In an external group, the next hop is computed with respect to the interface that is shared between the external peer and the local router.

## Multiple Instances of BGP

You can configure multiple instances of BGP at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Multiple instances of BGP are primarily used for Layer 3 VPN support.

IGP peers and external BGP (EBGP) peers (both nonmultihop and multihop) are all supported for routing instances. BGP peering is established over one of the interfaces configured under the **routing-instances** hierarchy.



**NOTE:** When a BGP neighbor sends BGP messages to the local routing device, the incoming interface on which these messages are received must be configured in the same routing instance that the BGP neighbor configuration exists in. This is true for neighbors that are a single hop away or multiple hops away.

Routes learned from the BGP peer are added to the **instance-name.inet.0** table by default. You can configure import and export policies to control the flow of information into and out of the instance routing table.

For Layer 3 VPN support, configure BGP on the provider edge (PE) router to receive routes from the customer edge (CE) router and to send the instances' routes to the CE router if necessary. You can use multiple instances of BGP to maintain separate per-site forwarding tables for keeping VPN traffic separate on the PE router.

You can configure import and export policies that allow the service provider to control and rate-limit traffic to and from the customer.

You can configure an EBGP multihop session for a VRF routing instance. Also, you can set up the EBGP peer between the PE and CE routers by using the loopback address of the CE router instead of the interface addresses.

- Related Documentation**
- [BGP Routes Overview on page 6](#)
  - [BGP Messages Overview on page 7](#)

---

## BGP Routes Overview

A BGP route is a destination, described as an IP address prefix, and information that describes the path to the destination.

The following information describes the path:

- AS path, which is a list of numbers of the ASs that a route passes through to reach the local router. The first number in the path is that of the last AS in the path—the AS closest to the local router. The last number in the path is the AS farthest from the local router, which is generally the origin of the path.
- Path attributes, which contain additional information about the AS path that is used in routing policy.

BGP peers advertise routes to each other in update messages.

BGP stores its routes in the Junos OS routing table (**inet.0**). The routing table stores the following information about BGP routes:

- Routing information learned from update messages received from peers
- Local routing information that BGP applies to routes because of local policies
- Information that BGP advertises to BGP peers in update messages

For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

The BGP router that first advertises a route assigns it one of the following values to identify its origin. During route selection, the lowest origin value is preferred.

- **0**—The router originally learned the route through an IGP (OSPF, IS-IS, or a static route).
- **1**—The router originally learned the route through an EGP (most likely BGP).
- **2**—The route's origin is unknown.

**Related  
Documentation**

- [Understanding BGP Path Selection on page 8](#)
- [Example: Advertising Multiple Paths in BGP on page 393](#)

---

## BGP Messages Overview

All BGP messages have the same fixed-size header, which contains a marker field that is used for both synchronization and authentication, a length field that indicates the length of the packet, and a type field that indicates the message type (for example, open, update, notification, keepalive, and so on).

This section discusses the following topics:

- [Open Messages on page 7](#)
- [Update Messages on page 8](#)
- [Keepalive Messages on page 8](#)
- [Notification Messages on page 8](#)

### Open Messages

After a TCP connection is established between two BGP systems, they exchange BGP open messages to create a BGP connection between them. Once the connection is established, the two systems can exchange BGP messages and data traffic.

Open messages consist of the BGP header plus the following fields:

- **Version**—The current BGP version number is 4.
- **Local AS number**—You configure this by including the **autonomous-system** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level.
- **Hold time**—Proposed hold-time value. You configure the local hold time with the BGP **hold-time** statement.

- BGP identifier—IP address of the BGP system. This address is determined when the system starts and is the same for every local interface and every BGP peer. You can configure the BGP identifier by including the **router-id** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level. By default, BGP uses the IP address of the first interface it finds in the router.
- Parameter field length and the parameter itself—These are optional fields.

## Update Messages

BGP systems send update messages to exchange network reachability information. BGP systems use this information to construct a graph that describes the relationships among all known ASs.

Update messages consist of the BGP header plus the following optional fields:

- Unfeasible routes length—Length of the withdrawn routes field
- Withdrawn routes—IP address prefixes for the routes being withdrawn from service because they are no longer deemed reachable
- Total path attribute length—Length of the path attributes field; it lists the path attributes for a feasible route to a destination
- Path attributes—Properties of the routes, including the path origin, the multiple exit discriminator (MED), the originating system's preference for the route, and information about aggregation, communities, confederations, and route reflection
- Network layer reachability information (NLRI)—IP address prefixes of feasible routes being advertised in the update message

## Keepalive Messages

BGP systems exchange keepalive messages to determine whether a link or host has failed or is no longer available. Keepalive messages are exchanged often enough so that the hold timer does not expire. These messages consist only of the BGP header.

## Notification Messages

BGP systems send notification messages when an error condition is detected. After the message is sent, the BGP session and the TCP connection between the BGP systems are closed. Notification messages consist of the BGP header plus the error code and subcode, and data that describes the error.

- Related Documentation**
- [Understanding BGP on page 4](#)
  - [BGP Routes Overview on page 6](#)

---

## Understanding BGP Path Selection

For each prefix in the routing table, the routing protocol process selects a single best path. After the best path is selected, the route is installed in the routing table. The best path becomes the active route if the same prefix is not learned by a protocol with a lower

(more preferred) global preference value, also known as the administrative distance. The algorithm for determining the active route is as follows:

1. Verify that the next hop can be resolved.
2. Choose the path with the lowest preference value (routing protocol process preference).

Routes that are not eligible to be used for forwarding (for example, because they were rejected by routing policy or because a next hop is inaccessible) have a preference of  $-1$  and are never chosen.

3. Prefer the path with higher local preference.

For non-BGP paths, choose the path with the lowest **preference2** value.

4. If the accumulated interior gateway protocol (AIGP) attribute is enabled, prefer the path with the lower AIGP attribute.
5. Prefer the path with the shortest autonomous system (AS) path value (skipped if the **as-path-ignore** statement is configured).

A confederation segment (sequence or set) has a path length of 0. An AS set has a path length of 1.

6. Prefer the route with the lower origin code.

Routes learned from an IGP have a lower origin code than those learned from an exterior gateway protocol (EGP), and both have lower origin codes than incomplete routes (routes whose origin is unknown).

7. Prefer the path with the lowest multiple exit discriminator (MED) metric.

Depending on whether nondeterministic routing table path selection behavior is configured, there are two possible cases:

- If nondeterministic routing table path selection behavior is not configured (that is, if the **path-selection cisco-nondeterministic** statement is not included in the BGP configuration), for paths with the same neighboring AS numbers at the front of the AS path, prefer the path with the lowest MED metric. To always compare MEDs whether or not the peer ASs of the compared routes are the same, include the **path-selection always-compare-med** statement.
- If nondeterministic routing table path selection behavior is configured (that is, the **path-selection cisco-nondeterministic** statement is included in the BGP configuration), prefer the path with the lowest MED metric.

Confederations are not considered when determining neighboring ASs. A missing MED metric is treated as if a MED were present but zero.



**NOTE:** MED comparison works for single path selection within an AS (when the route does not include an AS path), though this usage is uncommon.

By default, only the MEDs of routes that have the same peer autonomous systems (ASs) are compared. You can configure routing table path selection options to obtain different behaviors.

8. Prefer strictly internal paths, which include IGP routes and locally generated routes (static, direct, local, and so forth).
9. Prefer strictly external BGP (EBGP) paths over external paths learned through internal BGP (IBGP) sessions.
10. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.



**NOTE:** A path is considered a BGP equal-cost path (and will be used for forwarding) if a tie-break is performed after the previous step. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor, are considered.

BGP multipath does not apply to paths that share the same MED-plus-IGP cost yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

11. If both paths are external, prefer the currently active path to minimize route-flapping. This rule is not used if any one of the following conditions is true:
  - **path-selection external-router-id** is configured.
  - Both peers have the same router ID.
  - Either peer is a confederation peer.
  - Neither path is the current active path.
12. Prefer a primary route over a secondary route. A primary route is one that belongs to the routing table. A secondary route is one that is added to the routing table through an export policy.
13. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.
14. Prefer the path with the shortest cluster list length. The length is 0 for no list.
15. Prefer the path from the peer with the lowest peer IP address.

## Routing Table Path Selection

The shortest AS path step of the algorithm, by default, evaluates the length of the AS path and determines the active path. You can configure an option that enables Junos OS to skip this step of the algorithm by including the **as-path-ignore** option.



**NOTE:** The **as-path-ignore** option is not supported for routing instances.

To configure routing table path selection behavior, include the **path-selection** statement:

```

path-selection {
  (always-compare-med | cisco-non-deterministic | external-router-id);
  as-path-ignore;
  med-plus-igp {
    igp-multiplier number;
    med-multiplier number;
  }
}

```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Routing table path selection can be configured in one of the following ways:

- Emulate the Cisco IOS default behavior (**cisco-non-deterministic**). This mode evaluates routes in the order that they are received and does not group them according to their neighboring AS. With **cisco-non-deterministic** mode, the active path is always first. All inactive, but eligible, paths follow the active path and are maintained in the order in which they were received, with the most recent path first. Ineligible paths remain at the end of the list.

As an example, suppose you have three path advertisements for the 192.168.1.0 /24 route:

- Path 1—learned through EBGP; AS Path of 65010; MED of 200
- Path 2—learned through IBGP; AS Path of 65020; MED of 150; IGP cost of 5
- Path 3—learned through IBGP; AS Path of 65010; MED of 100; IGP cost of 10

These advertisements are received in quick succession, within a second, in the order listed. Path 3 is received most recently, so the routing device compares it against path 2, the next most recent advertisement. The cost to the IBGP peer is better for path 2, so the routing device eliminates path 3 from contention. When comparing paths 1 and 2, the routing device prefers path 1 because it is received from an EBGP peer. This allows the routing device to install path 1 as the active path for the route.



**NOTE:** We do not recommend using this configuration option in your network. It is provided solely for interoperability to allow all routing devices in the network to make consistent route selections.

- Always comparing MEDs whether or not the peer ASs of the compared routes are the same (**always-compare-med**).
- Override the rule that If both paths are external, the currently active path is preferred (**external-router-id**). Continue with the next step (Step 12) in the path-selection process.
- Adding the IGP cost to the next-hop destination to the MED value before comparing MED values for path selection (**med-plus-igp**).

BGP multipath does not apply to paths that share the same MED-plus-IGP cost, yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

## Effects of Advertising Multiple Paths to a Destination

BGP advertises only the active path, unless you configure BGP to advertise multiple paths to a destination.

Suppose a routing device has in its routing table four paths to a destination and is configured to advertise up to three paths (**add-path send path-count 3**). The three paths are chosen based on path selection criteria. That is, the three best paths are chosen in path-selection order. The best path is the active path. This path is removed from consideration and a new best path is chosen. This process is repeated until the specified number of paths is reached.

### Related Documentation

- [Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 260](#)
- [Examples: Configuring BGP MED on page 92](#)
- [Example: Advertising Multiple BGP Paths to a Destination on page 392](#)

---

## BGP Configuration Overview

To configure the device as a node in a BGP network:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Configure point-to-point peering sessions. See [“Example: Configuring External BGP Point-to-Point Peer Sessions” on page 34](#).
3. Configure IBGP sessions between peers. See [“Example: Configuring Internal BGP Peer Sessions” on page 57](#).
4. Configure a routing policy to advertise the BGP routes.
5. (Optional) Configure route reflector clusters. See [“Example: Configuring a Route Reflector” on page 421](#).
6. (Optional) Subdivide autonomous systems (ASs). See [“Example: Configuring BGP Confederations” on page 442](#).
7. (Optional) Assign a router ID to each routing device running BGP.
8. (Optional) Configure a local preference to direct all outbound AS traffic to a specific peer. See [“Example: Configuring the Local Preference Value for BGP Routes” on page 80](#) in the *Junos OS Routing Protocols Library for Routing Devices*.
9. (Optional) Configure routing table path selection options that define different ways to compare multiple exit discriminators (MEDs). See [“Understanding BGP Path Selection” on page 8](#) in the *Junos OS Routing Protocols Library for Routing Devices*.

### Related Documentation

- [Understanding BGP on page 4](#)

## CHAPTER 2

# BGP Standards

- [Supported BGP Standards on page 13](#)

### Supported BGP Standards

---

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for IP version 4 (IPv4) BGP.

For a list of supported IP version 6 (IPv6) BGP standards, see *Supported IPv6 Standards*.

Junos OS BGP supports authentication for protocol exchanges (MD5 authentication).

- RFC 1745, *BGP4/IDRP for IP—OSPF Interaction*
- RFC 1772, *Application of the Border Gateway Protocol in the Internet*
- RFC 1997, *BGP Communities Attribute*
- RFC 2283, *Multiprotocol Extensions for BGP-4*
- RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*
- RFC 2439, *BGP Route Flap Damping*
- RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
- RFC 2796, *BGP Route Reflection – An Alternative to Full Mesh IBGP*
- RFC 2858, *Multiprotocol Extensions for BGP-4*
- RFC 2918, *Route Refresh Capability for BGP-4*
- RFC 3065, *Autonomous System Confederations for BGP*
- RFC 3107, *Carrying Label Information in BGP-4*
- RFC 3345, *Border Gateway Protocol (BGP) Persistent Route Oscillation Condition*
- RFC 3392, *Capabilities Advertisement with BGP-4*
- RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*
- RFC 4273, *Definitions of Managed Objects for BGP-4*
- RFC 4360, *BGP Extended Communities Attribute*
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

- RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*
- RFC 4486, *Subcodes for BGP Cease Notification Message*
- RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
- RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*
- RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*
- RFC 4724, *Graceful Restart Mechanism for BGP*
- RFC 4760, *Multiprotocol Extensions for BGP-4*
- RFC 4781, *Graceful Restart Mechanism for BGP with MPLS*
- RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

Option 4b (eBGP redistribution of labeled IPv6 routes from AS to neighboring AS) is not supported.

- RFC 4893, *BGP Support for Four-octet AS Number Space*
- RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*
- RFC 5065, *Autonomous System Confederations for BGP*
- RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*
- RFC 5291, *Outbound Route Filtering Capability for BGP-4 (partial support)*
- RFC 5292, *Address-Prefix-Based Outbound Route Filter for BGP-4 (partial support)*

Devices running Junos OS can receive prefix-based ORF messages.

- RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*
- RFC 5492, *Capabilities Advertisement with BGP-4*
- RFC 5575, *Dissemination of flow specification rules*
- RFC 5668, *4-Octet AS Specific BGP Extended Community*
- RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*
- RFC 6811, *BGP Prefix Origin Validation*
- RFC 7311, *The Accumulated IGP Metric Attribute for BGP*
- Internet draft draft-ietf-idr-add-paths-06.txt, *Advertisement of Multiple Paths in BGP* (expires March 2012)
- Internet draft draft-ietf-idr-link-bandwidth-01.txt, *BGP Link Bandwidth Extended Community* (expires August 2010)
- Internet draft draft-ietf-sidr-origin-validation-signaling-00, *BGP Prefix Origin Validation State Extended Community (partial support)* (expires May 2011)

The extended community (origin validation state) is supported in Junos OS routing policy. The specified change in the route selection procedure is not supported.

- Internet draft draft-kato-bgp-ipv6-link-local-00.txt, *BGP4+ Peering Using IPv6 Link-local Address*

The following RFCs and Internet draft do not define standards, but provide information about BGP and related technologies. The IETF classifies them variously as “Experimental” or “Informational.”

- RFC 1965, *Autonomous System Confederations for BGP*
- RFC 1966, *BGP Route Reflection—An alternative to full mesh IBGP*
- RFC 2270, *Using a Dedicated AS for Sites Homed to a Single Provider*
- Internet draft draft-ietf-ngtrans-bgp-tunnel-04.txt, *Connecting IPv6 Islands across IPv4 Clouds with BGP* (expires July 2002)

**Related  
Documentation**

- *Supported IPv6 Standards*
- *Accessing Standards Documents on the Internet*



## CHAPTER 3

# Understanding 4-Byte Autonomous System Numbers

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)

### 4-Byte Autonomous System Numbers Overview

---

This Technology Overview describes 4-byte autonomous system (AS) numbers and the operation of BGP in a network with a mix of 2-byte and 4-byte AS numbers.

The 2-byte AS number, also known as a 16-bit AS number or 2-octet AS number, provides a pool of 65,536 AS numbers. The 2-byte AS number range has been exhausted. 4-byte AS numbers are specified in RFC 4893, *BGP Support for Four-Octet AS Number Space* and provide a pool of 4,294,967,296 AS numbers.

As of January 1, 2009 the Internet Assigned Numbers Authority (IANA) only assigns 4-byte AS numbers, unless a 2-byte AS number is specifically requested. The Internet Engineering Task Force (IETF) RFC 4893 defines a method for smooth transition from 2-byte AS numbers to 4-byte AS numbers and for maintaining backward compatibility.

RFC 4893 introduces two new optional transitive BGP attributes, AS4\_PATH and AS4\_AGGREGATOR. These new attributes are used to propagate 4-byte AS path information across BGP speakers that do not support 4-byte AS numbers.

RFC 4893 also introduces a reserved, well-known, 2-byte AS number, AS 23456. This reserved AS number is called AS\_TRANS in RFC 4893.

This Technology Overview presents the following topics:

- Juniper Networks implementation of 4-byte AS Numbers
- How to configure 4-byte AS numbers
- The impact on BGP peering sessions with a mix of 2-byte and 4-byte AS numbers
- The impact on BGP paths with a mix of 2-byte and 4-byte AS numbers
- How to configure route distinguishers using 4-byte AS numbers
- How to configure extended community attributes using 4-byte AS numbers
- The impact on BGP route loop detection with a mix of 2-byte and 4-byte AS numbers

**Related  
Documentation**

- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

---

## Juniper Networks Implementation of 4-Byte Autonomous System Numbers

Junos OS Release 9.1 and later supports 4-byte AS numbers.

If your network is currently using 2-byte AS numbers, you are not required to get new 4-byte AS numbers. The 2-byte AS number range is a subset of the 4-byte AS number range. A Juniper networks router that supports 4-byte AS numbers simply prepends a string of zeros in front of the 2-byte AS number. For example, the 2-byte AS number 65000 becomes the 4-byte AS number 00000.65000.

If your Juniper Networks router supports 4-byte AS numbers and has a peer relationship with a router that does not support 4-byte AS numbers, the following sequence takes place in the adjacent RIB-in routing table after the router that supports 4-byte AS numbers advertises this capability to the new peer:

1. The router that supports 4-byte AS numbers receives an advertisement from the peer that supports only 2-byte AS numbers.
2. On the router that supports 4-byte AS numbers, the 2-byte AS path is converted into the 4-byte AS number by prepending a string of zeros in front of the 2-byte AS number.
3. If a 4-byte AS number is also present in the path, it is merged with the 2-byte AS numbers in the path.
4. If the AGGREGATOR and AS4\_AGGREGATOR attributes are present, these attributes are also merged.

If your Juniper Networks router supports 4-byte AS numbers and has a peer relationship with a router that does not support 4-byte AS numbers, the following sequence takes place in the adjacent RIB-out routing table:

1. Update message are reformatted before being sent to the router that does not support 4-byte AS numbers.
2. The router that supports 4-byte AS numbers sends the 4-byte AS number in the AS4\_PATH attribute.
3. The AS\_PATH attribute is also sent. It is encoded with the 2-byte AS numbers. Mappable 4-byte AS numbers, below 64537, are sent as 2-byte AS numbers. Non-mappable 4-byte AS numbers, above 64536, are represented by the well-known 2-byte AS number, AS 23456.
4. A single peer group is used for the routers that support 4-byte AS numbers and the routers that support only 2-byte AS numbers.

**Related Documentation**

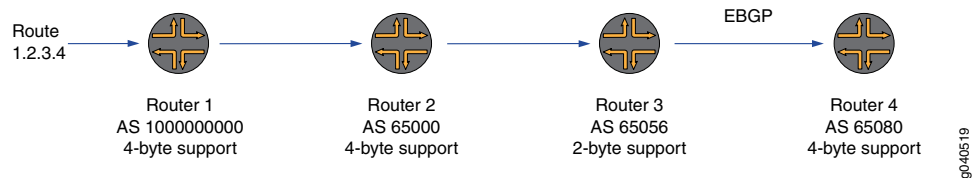
- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

## Prepending 4-Byte AS Numbers in an AS Path

This section describes how to prepend one or more AS numbers at the beginning of an AS path. The AS numbers are added at the beginning of the path after the actual AS number from which the route originates has been added to the path. Prepending an AS path makes a shorter AS path look longer and therefore less preferable to BGP.

In [Figure 2 on page 20](#), Router 2 is configured to prepend AS 1000000000 4 times in front of AS number 65000.

**Figure 2: EBGP With 4-Byte AS Numbers Prepend to the AS Path**



You can display the route details using the `show route` command on Router 3. In the following example, notice that the prepended AS number displayed in the AS path on Router 3 is the AS\_TRANS number, AS 23456. This is because Router 3 does not support 4-byte AS numbers.

```
user@Router3# show route 1.2.3.4 detail
...
1.2.3.4/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.3
                  AS path: 65000 23456 23456 23456 23456 I
```

You can display the route details using the `show route` command on Router 4. In the following example, notice that the prepended AS number displayed in the AS path on Router 4 is AS 1000000000. This is because Router 4 supports 4-byte AS numbers and merges the AS\_PATH and AS4\_PATH attributes.

```
user@Router4# show route 1.2.3.4 detail
...
1.2.3.4/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.9
                  AS path: 65056 65000 1000000000 1000000000 1000000000 1000000000 I
```

- Related Documentation**
- [4-Byte Autonomous System Numbers Overview on page 17](#)
  - [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
  - [Configuring 4-Byte Autonomous System Numbers on page 208](#)
  - [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
  - [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
  - [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
  - [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
  - [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)

- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

## Configuring 4-Byte AS Numbers and BGP Extended Community Attributes

A BGP community is a group of destinations that share a common property. You can configure the standard community attribute and extended community attributes for inclusion in BGP update messages.

For example, when configuring a VPN routing and forwarding (VRF) instance, you need to configure a route target. A route target is one type of BGP extended community attribute. To create a named BGP extended community attribute, include the **community** statement and specify the community members:

```
community name {
  members [ community-ids ];
}
```

To specify the community members, you must specify the community ID. The community ID consists of three components that you specify in the following format:

*type:administrator:assigned-number*

The **administrator** field of some BGP extended community attributes is an AS number. To configure a **target** extended community, which includes a 4-byte AS number in the plain-number format, append the letter “L” to the end of the number.

In the following example, a **target** community with the 4-byte AS number **334324** and an assigned number of **132** is represented as **target:334324L:132**.

```
[edit policy-options]
community vpn_blue members [ target:334324L:132 ];
```



**NOTE:** If you display the target extended community information on a peer router that does not support 4-byte AS numbers, the router displays **target:unknown** format.

### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)

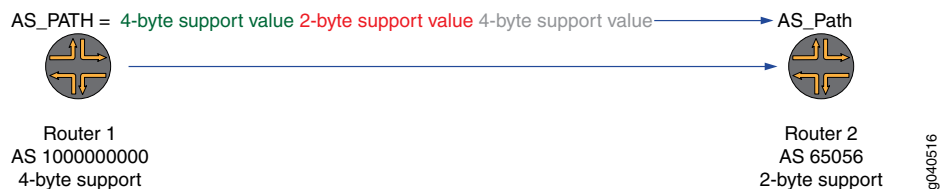
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

## Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain

This section describes what happens when a router that supports 4-byte AS numbers sends the AS path statement to a router that only supports 2-byte AS numbers if the first router is configured with an AS number outside the 2-byte AS number range.

In [Figure 3 on page 22](#) Router 1 supports 4-byte AS numbers. Router 1 is configured to use a 4-byte AS number, AS 1000000000. Router 2 supports 2-byte AS numbers. Router 2 is configured with a 2-byte AS number, AS 65056.

**Figure 3: 4-Byte Capable Router AS Path to a 2-Byte Capable Router**

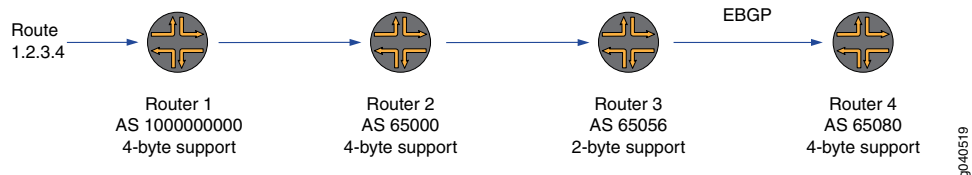


- Router 2 does not accept 4-byte AS numbers in the AS\_PATH attribute. You can verify this using the **show bgp neighbor** command on Router 1.

```
user@Router1# show bgp neighbor 192.168.1.9 | match "AS"
Peer: 192.168.1.9+179 AS 65056 Local: 192.168.1.2+64053 AS 65080
Peer does not support 4 byte AS extension
```

[Figure 4 on page 22](#) shows four routers running EBGp. Router 1, Router 2, and Router 4 support 4-byte AS numbers. Router 3 does not support 4-byte AS numbers.

**Figure 4: EBGp 4-Byte AS Path Through a 2-Byte AS Domain**



In this case:

- Router 1 sends the 4-byte AS number, AS 1000000000, in the AS\_PATH attribute to Router 2.
- Router 2 knows that Router 3 does not support 4-byte AS numbers.
- Router 2 sends the AS\_TRANS number, AS 23456, in the AS\_PATH attribute in place of the 4-byte AS number to Router 3.
- Router 2 sends the 4-byte AS number, AS 1000000000 in the AS4\_PATH attribute to Router 3.

- Because the AS4\_PATH attribute is transitive, Router 3 sends both the AS\_PATH attribute and the AS4\_PATH attribute to Router 4.
- When Router 4 receives the AS\_PATH and AS4\_PATH attributes, it merges the path statements to create an accurate AS path.

You can display the AS path using the **show route** command on Router 3. In the following example, notice that the AS number 23456 appears in the AS path and that the AS4\_PATH attribute is **Unrecognized**. Because the AS4\_PATH attribute is a transitive attribute, it is forwarded to the next router.

```
user@Router3# show route 1.2.3.4 detail
AS path: 65000 23456 I Unrecognized Attributes: 13 bytes
```

You can display the route details using the **show route** command on Router 4. In the following example, notice that as the AS path transitions Router 3, as shown in the AS2 (2-byte AS) path, the AS number is displayed as AS\_TRANS. This means that Router 3 sees the AS number as 23456. In the AS4 (4-byte AS) path the AS number is displayed as 1000000000. In the merged AS path the correct AS path numbers are displayed for AS 65056, AS 65000, and AS 1000000000.

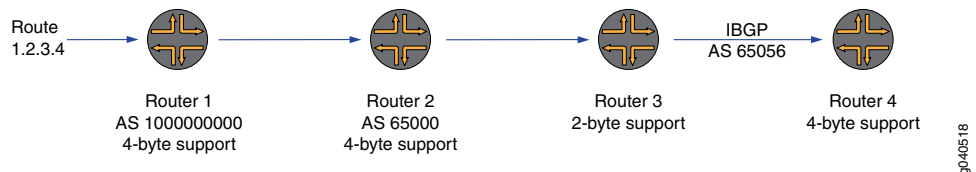
```
user@Router4# show route 1.2.3.4 detail
...
AS path: AS2 PA[3]:65056 65000 AS_TRANS

AS path: AS4 PA[2]:65056 1000000000

AS path: Merged[3]:65056 65000 1000000000 I
```

Figure 5 on page 23 shows 4 routers running IBGP. Router 1, Router 2, and Router 4 support 4-byte AS numbers. Router 3 does not support 4-byte AS numbers.

**Figure 5: IBGP 4-Byte AS Path Through a 2-Byte AS Domain**



In this case:

- Router 1 sends the 4-byte AS number, AS 1000000000, in the AS\_PATH attribute to Router 2.
- Router 2 knows that Router 3 does not support 4-byte AS numbers.
- Router 2 sends the AS\_TRANS number, AS 23456, in the AS\_PATH attribute in place of the 4-byte AS number to Router 3.
- Router 3 sends both the AS\_PATH attribute and the AS4\_PATH attribute to Router 4.
- When Router 4 receives the AS\_PATH and AS4\_PATH attributes, it merges the path statements to create an accurate AS path.

You can display the route details using the **show route** command on Router 2. In the following example, notice that the AS path is displayed as 10000000000.

```
user@Router2# show route 1.2.3.4 detail
...
AS path: 10000000000
```

You can display the route details using the **show route** command on Router 3. In the following example, notice that the AS path is displayed as 65000 23456.

```
user@Router3# show route 1.2.3.4 detail
...
AS path: 65000 23456 I
```

You can display the route details using the **show route** command on Router 4. In the following example, notice that the merged AS path is displayed as 65000 10000000000.

```
user@Router4# show route 1.2.3.4 detail
...
AS path: 65000 10000000000 I
```

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)

---

## Understanding 4-Byte AS Numbers and Route Distinguishers

A route distinguisher (RD) is an 8-byte field prefixed to a service provider customer's IPv4 address. The resulting 12-byte field is a unique VPN-IPv4 address. The RD in BGP messages consists of two major fields, the type field (2 bytes) and value field (6 bytes). The type field determines how the value field should be interpreted.

The route distinguisher is configured as a 6-byte value that you can specify as **as-number:number**, where **as-number** is your assigned AS number and **number** (also known as an administrative number or assigned number subfield) is any 2-byte or 4-byte value. The AS number can be in the range from 1 through 4,294,967,295. If the AS number is a 2-byte value, the administrative number is a 4-byte value. If the AS number is 4-byte value, the administrative number is a 2-byte value.

An RD consisting of a 4-byte AS number and a 2-byte administrative number is defined as a type 2 route distinguisher in RFC 4364, *BGP/MPLS IP Virtual Private Networks*.

To configure an RD using a 4-byte AS number, append the letter "L" to the end of the number. In the following example, the 4-byte AS number is 7765000 and the administrative number is 1000:

```
user@Router1# set routing-instances 4B route-distinguisher 7765000L:1000
```

If the router you are configuring is a BGP peer of a router that does not support 4-byte AS numbers, you also need to configure a local AS number as discussed in [“Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number” on page 28](#). To configure the local AS number, include the **local-as** statement, specify the 2-byte AS number to use (65001), and include the **private** option.

```
user@Router1# set routing-instances 4B protocols bgp group 4B2Bpeers local-as 65001
private
```

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

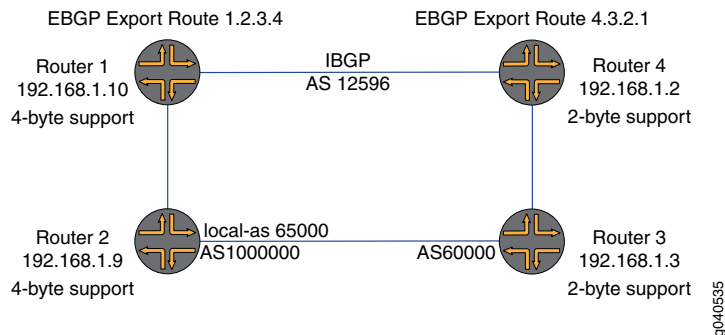
---

## Understanding 4-Byte AS Numbers and Route Loop Detection

One of the most important functions in BGP is route loop detection at the autonomous system level using the AS\_PATH attribute. A simple way of thinking of the AS\_PATH is that it is the list of autonomous systems that a route goes through to reach its destination. Loops are detected and avoided by the router checking for its own AS number in the AS\_PATH received from a neighboring AS.

This section describes how route loop detection works with a mix of routers that support and do not support 4-byte AS numbers. [Figure 6 on page 26](#) shows a small network with the potential for BGP loops.

**Figure 6: 4-Byte AS Numbers and Loop Detection**



In the first example, an EBGP route, route 1.2.3.4, is first advertised by Router 1. The first AS in the path is AS 12596 as configured on Router 1. The second AS that is in the path is AS 1000000 as configured on Router 2. AS 1000000 is sent in the AS4\_path attribute and the AS\_TRANS number, AS 23456, is sent in the AS\_PATH attribute to Router 3. The third AS that is in the path is AS 60000, as configured on Router 3.

The **show route** command output shows the AS path for route 1.2.3.4 as advertised by Router 3 to Router 4. In the **show route** command output, you see AS 12596 first. Because Router 3 does not support 4-byte AS numbers, you see AS 23456 second. Because Router 2 used a local AS of 65000 to establish a peer relationship with Router 3, you see AS 65000 third. AS 60000 is not in the **show route** command output because the command was entered on the router configured with AS 60000.

```
user@Router3# show route advertising-protocol bgp 192.168.1.2
...
Prefix Nexthop MED Lc1pref AS path
10.255.14.172/32 Self 65000 23456 12596 I
```

In this case, when Router 4 sees its own AS number, AS 12596, in the path, it detects a routing loop.

In the second example, an EBGP route, route 4.3.2.1, is first advertised by Router 4. The first AS in the path is AS 12596 as configured on Router 4. The second AS in the path is AS 60000 as configured on Router 3. The third AS is in the path is AS 1000000 as configured on Router 2.

The **show route** command output shows the AS path for route 4.3.2.1 as advertised by Router 2 to Router 1. In the **show route** command output, you see AS 12596 first and AS 60000 second. AS 1000000 is not in the **show route** command output because the command was entered on the router configured with AS 1000000.

```
user@Router2# show route advertising-protocol bgp 192.168.1.10
...
Prefix Nexthop MED Lc1pref AS path
10.255.14.172/32 Self 60000 12596 I
```

When Router 1 sees its own AS number, AS 12596, in the path, it detects a routing loop.

**Related Documentation**

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)
- [Disabling Attribute Set Messages on Independent AS Domains for BGP Loop Detection on page 209](#)

## Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number

This section describes what happens when a router that supports 4-byte AS numbers establishes a peer relationship with a router that only supports 2-byte AS numbers if both routers are configured with AS numbers in the 2-byte AS number range.

In [Figure 7 on page 27](#), Router 1 is running Junos OS Release 9.2 that supports 4-byte AS numbers. Router 1 is configured to use a 2-byte AS number, AS 12596. Router 2 is running Junos OS Release 8.5 that supports 2-byte AS numbers. Router 2 is configured with a 2-byte AS number, AS 60000.

**Figure 7: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 2-Byte AS Number**



- The following example shows the relevant portion of the Router 1 configuration.

```

user@Router1# show configuration
...
autonomous-system 12596;
...
local-address 192.168.1.10;
export static-to-bgp;
peer-as 60000;
  
```

- To verify that the AS path of route 1.2.3.4 contains AS 12596, use the **show route** command on Router 2. The following example shows that the BGP peer session is established in the normal way and that the AS path of route 1.2.3.4 contains AS 12596:

```
user@Router2# show route 1.2.3.4
1.2.3.4/32          *[BGP/170] 00:01:29, localpref 100, from 192.168.1.10
                    AS path: 12596 I
                    > via at-0/1/0.1001
```

- To display the session-establishment messages logged on Router 1, use the **show log messages** command. The following example shows that Router 1 discovers that Router 2 does not support 4-byte AS numbers:

```
user@Router1# show log messages
Nov  7 09:41:39.443493 bgp_4byte_aspath_add_cap():153 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): 4 byte AS capability added, AS 12596
Nov  7 09:41:39.443582 bgp_send: sending 67 bytes to 192.168.1.9 (External AS
60000)
[...]
Nov  7 09:41:39.448055 bgp_4byte_aspath_adjust():1279 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): Adjust BGP update to Old/New BGP speaker format
Nov  7 09:41:39.448132 bgp_4byte_aspath_adjust():1290 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): Cached information of previous update format is not
used
Nov  7 09:41:39.448162 bgp_generate_2byte_aspath():422 AS4-Peer 192.168.1.9
(External AS 60000)(SEND): Generating 2 byte AS path from 4 byte as-path
Nov  7 09:41:39.448198 bgp_send: sending 64 bytes to 192.168.1.9 (External AS
60000)
```

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

## Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number

This section describes what happens when a router that supports 4-byte AS numbers establishes a peer relationship with a router that only supports 2-byte AS numbers if the first router is configured with an AS number outside the 2-byte AS number range.

In [Figure 8 on page 29](#), Router 2 is running Junos OS Release 9.2 that supports 4-byte AS numbers. Router 2 is configured to use a 4-byte AS number, AS 1000000. Router 3 is

running Junos OS Release 8.5 that supports 2-byte AS numbers. Router 3 is configured with a 2-byte AS number, AS 60000.

**Figure 8: 4-Byte Capable Router Having a Peer Relationship With a 2-Byte Capable Router Using a 4-Byte AS Number**



You can configure a local AS number to be used only during the establishment of the BGP session with a BGP neighbor, but to be hidden in the AS path sent to external BGP peers. To configure the local AS number, include the **local-as** statement, specify the 2-byte AS number to use, 65530, and include the **private** option. With this configuration, only the global AS number, 1000000, is included in the AS path sent to external peers. The following example shows the relevant portion of the Router 1 configuration:

```

user@Router1# show configuration
...
autonomous-system 1000000;
...
local-address 192.168.1.9;
export static-to-bgp;
neighbor 192.168.1.3 {
    peer-as 60000;
    local-as 65530 private;
}
  
```

The peer AS number on Router 3 should equal the local AS number on Router 1. The following example shows the relevant portion of the Router 3 configuration:

```

user@Router3# show configuration
...
autonomous-system 60000;
...
local-address 192.168.1.3;
neighbor 192.168.1.9 {
    peer-as 65530;
}
  
```

To verify that the AS path of route 22.1.2.3 contains AS 65530, use the **show route** command on Router 3. The following example shows that the BGP peer session is established and that the AS path of route 22.1.2.3 contains AS 65530:

```

user@Router3# show route 22.1.2.3
...
22.1.2.3/32      *[BGP/170] 01:39:55, localpref 100, from 192.168.1.9
                  AS path: 65530 I
                  > via so-1/0/3.0
  
```

#### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)

- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)
- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)

## PART 2

# Configuration

- [Basic BGP Configuration on page 33](#)
- [BGP Path Attribute Configuration on page 79](#)
- [BGP Policy Configuration on page 211](#)
- [BGP BFD Configuration on page 349](#)
- [BGP Load Balancing Configuration on page 365](#)
- [IBGP Scaling Configuration on page 419](#)
- [BGP Security Configuration on page 449](#)
- [BGP Flap Configuration on page 497](#)
- [Multiprotocol BGP Configuration on page 537](#)
- [BGP CLNS Configuration on page 569](#)
- [BGP Monitoring Configuration on page 577](#)
- [BGP Configuration Statements on page 593](#)



## CHAPTER 4

# Basic BGP Configuration

- [Examples: Configuring External BGP Peering on page 33](#)
- [Examples: Configuring Internal BGP Peering on page 56](#)

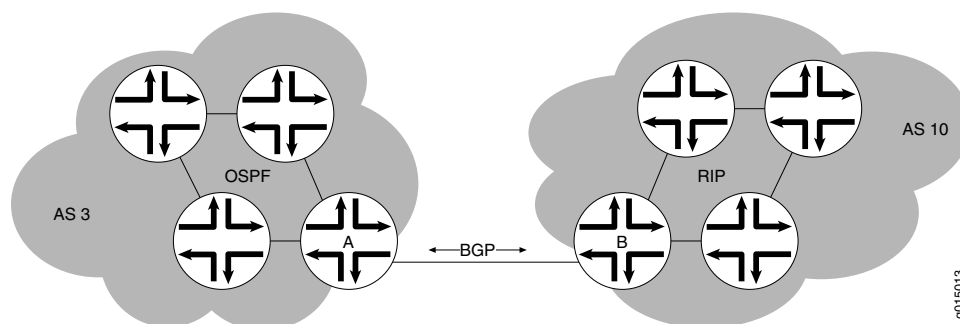
## Examples: Configuring External BGP Peering

- [Understanding External BGP Peering Sessions on page 33](#)
- [Example: Configuring External BGP Point-to-Point Peer Sessions on page 34](#)
- [Example: Configuring External BGP on Logical Systems with IPv6 Interfaces on page 41](#)

## Understanding External BGP Peering Sessions

To establish point-to-point connections between peer autonomous systems (ASs), you configure a BGP session on each interface of a point-to-point link. Generally, such sessions are made at network exit points with neighboring hosts outside the AS. [Figure 9 on page 33](#) shows an example of a BGP peering session.

Figure 9: BGP Peering Session



In [Figure 9 on page 33](#), Router A is a gateway router for AS 3, and Router B is a gateway router for AS 10. For traffic internal to either AS, an interior gateway protocol (IGP) is used (OSPF, for instance). To route traffic between peer ASs, a BGP session is used.

You arrange BGP routing devices into groups of peers. Different peer groups can have different group types, AS numbers, and route reflector cluster identifiers.

To define a BGP group that recognizes only the specified BGP systems as peers, statically configure all the system's peers by including one or more **neighbor** statements. The peer neighbor's address can be either an IPv6 or IPv4 address.

As the number of external BGP (EBGP) groups increases, the ability to support a large number of BGP sessions might become a scaling issue. The preferred way to configure a large number of BGP neighbors is to configure a few groups consisting of multiple neighbors per group. Supporting fewer EBGP groups generally scales better than supporting a large number of EBGP groups. This becomes more evident in the case of hundreds of EBGP groups when compared with a few EBGP groups with multiple peers in each group.

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP RIB and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

### Example: Configuring External BGP Point-to-Point Peer Sessions

This example shows how to configure BGP point-to-point peer sessions.

- [Requirements on page 34](#)
- [Overview on page 34](#)
- [Configuration on page 35](#)
- [Verification on page 37](#)

#### Requirements

---

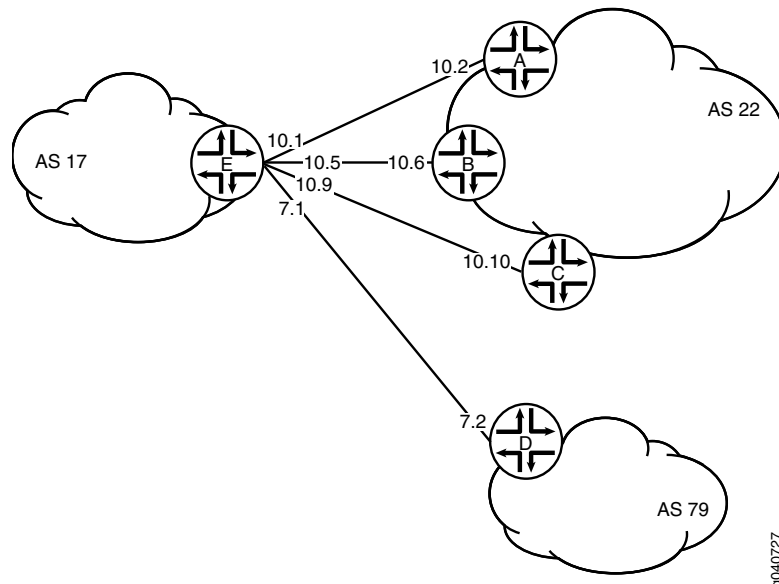
Before you begin, if the default BGP policy is not adequate for your network, configure routing policies to filter incoming BGP routes and to advertise BGP routes.

#### Overview

---

[Figure 10 on page 35](#) shows a network with BGP peer sessions. In the sample network, Device E in AS 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22 and have IP addresses 10.10.10.2, 10.10.10.6, and 10.10.10.10. Peer D resides in AS 79, at IP address 10.21.7.2. This example shows the configuration on Device E.

Figure 10: Typical Network with BGP Peer Sessions



9040727

### 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.

```
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-0/0/1 unit 5 description to-B
set interfaces ge-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-0/1/0 unit 9 description to-C
set interfaces ge-0/1/0 unit 9 family inet address 10.10.10.9/30
set interfaces ge-1/2/1 unit 21 description to-D
set interfaces ge-1/2/1 unit 21 family inet address 10.21.7.1/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set protocols bgp group external-peers neighbor 10.21.7.2 peer-as 79
set routing-options autonomous-system 17
```

**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 BGP peer sessions:

1. Configure the interfaces to Peers A, B, C, and D.

[edit interfaces]

```

user@E# set ge-1/2/0 unit 0 description to-A
user@E# set ge-1/2/0 unit 0 family inet address 10.10.10.1/30
user@E# set ge-0/0/1 unit 5 description to-B
user@E# set ge-0/0/1 unit 5 family inet address 10.10.10.5/30
user@E# set ge-0/1/0 unit 9 description to-C
user@E# set ge-0/1/0 unit 9 family inet address 10.10.10.9/30
user@E# set ge-1/2/1 unit 21 description to-D
user@E# set ge-1/2/1 unit 21 family inet address 10.21.7.1/30

```

2. Set the autonomous system (AS) number.

```

[edit routing-options]
user@E# set autonomous-system 17

```

3. Create the BGP group, and add the external neighbor addresses.

```

[edit protocols bgp group external-peers]
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10

```

4. Specify the autonomous system (AS) number of the external AS.

```

[edit protocols bgp group external-peers]
user@E# set peer-as 22

```

5. Add Peer D, and set the AS number at the individual neighbor level.

The neighbor configuration overrides the group configuration. So, while **peer-as 22** is set for all the other neighbors in the group, **peer-as 79** is set for neighbor 10.21.7.2.

```

[edit protocols bgp group external-peers]
user@E# set neighbor 10.21.7.2 peer-as 79

```

6. Set the peer type to external BGP (EBGP).

```

[edit protocols bgp group external-peers]
user@E# set type external

```

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

```

[edit]
user@E# show interfaces
ge-1/2/0 {
  unit 0 {
    description to-A;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
ge-0/0/1 {
  unit 5 {
    description to-B;
    family inet {
      address 10.10.10.5/30;
    }
  }
}

```

```

    }
  }
  ge-0/1/0 {
    unit 9 {
      description to-C;
      family inet {
        address 10.10.10.9/30;
      }
    }
  }
  ge-1/2/1 {
    unit 21 {
      description to-D;
      family inet {
        address 10.21.7.1/30;
      }
    }
  }
}

[edit]
user@E# show protocols
bgp {
  group external-peers {
    type external;
    peer-as 22;
    neighbor 10.10.10.2;
    neighbor 10.10.10.6;
    neighbor 10.10.10.10;
    neighbor 10.21.7.2 {
      peer-as 79;
    }
  }
}

[edit]
user@E# show routing-options
autonomous-system 17;

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 37](#)
- [Verifying BGP Groups on page 40](#)
- [Verifying BGP Summary Information on page 40](#)

### Verifying BGP Neighbors

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

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

```
user@E> show bgp neighbor
```

```

Peer: 10.10.10.2+179 AS 22      Local: 10.10.10.1+65406 AS 17
Type: External      State: Established      Flags: <Sync>
Last State: OpenConfirm      Last Event: RecvKeepAlive
Last Error: None
Options: <Preference PeerAS Refresh>
Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.10.10.2      Local ID: 10.10.10.1      Active Holdtime: 90
Keepalive Interval: 30      Peer index: 0
BFD: disabled, down
Local Interface: ge-1/2/0.0
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 22)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      0
Last traffic (seconds): Received 10      Sent 6      Checked 1
Input messages: Total 8522      Updates 1      Refreshes 0      Octets 161922
Output messages: Total 8433      Updates 0      Refreshes 0      Octets 160290
Output Queue[0]: 0

Peer: 10.10.10.6+54781 AS 22      Local: 10.10.10.5+179 AS 17
Type: External      State: Established      Flags: <Sync>
Last State: OpenConfirm      Last Event: RecvKeepAlive
Last Error: None
Options: <Preference PeerAS Refresh>
Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.10.10.6      Local ID: 10.10.10.1      Active Holdtime: 90
Keepalive Interval: 30      Peer index: 1
BFD: disabled, down
Local Interface: ge-0/0/1.5
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 22)
Peer does not support Addpath

```

```

Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      0
Last traffic (seconds): Received 12   Sent 6   Checked 33
Input messages:  Total 8527   Updates 1   Refreshes 0   Octets 162057
Output messages: Total 8430   Updates 0   Refreshes 0   Octets 160233
Output Queue[0]: 0

Peer: 10.10.10.10+55012 AS 22  Local: 10.10.10.9+179 AS 17
Type: External  State: Established  Flags: <Sync>
Last State: OpenConfirm  Last Event: RecvKeepAlive
Last Error: None
Options: <Preference PeerAS Refresh>
Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.10.10.10      Local ID: 10.10.10.1      Active Holdtime: 90
Keepalive Interval: 30    Peer index: 2
BFD: disabled, down
Local Interface: fe-0/1/0.9
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 22)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      0
Last traffic (seconds): Received 15   Sent 6   Checked 37
Input messages:  Total 8527   Updates 1   Refreshes 0   Octets 162057
Output messages: Total 8429   Updates 0   Refreshes 0   Octets 160214
Output Queue[0]: 0

Peer: 10.21.7.2+61867 AS 79  Local: 10.21.7.1+179 AS 17
Type: External  State: Established  Flags: <ImportEval Sync>
Last State: OpenConfirm  Last Event: RecvKeepAlive
Last Error: None
Options: <Preference PeerAS Refresh>
Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.21.7.2      Local ID: 10.10.10.1      Active Holdtime: 90
Keepalive Interval: 30    Peer index: 3
BFD: disabled, down
Local Interface: ge-1/2/1.21

```

```

NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 79)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      0
Last traffic (seconds): Received 28   Sent 24   Checked 47
Input messages: Total 8521   Updates 1   Refreshes 0   Octets 161943
Output messages: Total 8427   Updates 0   Refreshes 0   Octets 160176
Output Queue[0]: 0

```

### Verifying BGP Groups

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From operational mode, run the **show bgp group** command.

```

user@E> show bgp group
Group Type: External                               Local AS: 17
Name: external-peers  Index: 0                     Flags: <>
Holdtime: 0
Total peers: 4      Established: 4
10.10.10.2+179
10.10.10.6+54781
10.10.10.10+55012
10.21.7.2+61867
inet.0: 0/0/0/0

Groups: 1  Peers: 4   External: 4   Internal: 0   Down peers: 0   Flaps: 0
Table      Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet.0      0          0          0          0          0          0          0

```

### Verifying BGP Summary Information

**Purpose** Verify that the BGP configuration is correct.

**Action** From operational mode, run the **show bgp summary** command.

```

user@E> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet.0      0          0          0          0          0          0          0
Peer      AS      InPkt    OutPkt    OutQ    Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.10.10.2      22      8559      8470      0        0 2d 16:12:56

```

0/0/0/0	0/0/0/0				
10.10.10.6	22	8566	8468	0	0 2d 16:12:12
0/0/0/0	0/0/0/0				
10.10.10.10	22	8565	8466	0	0 2d 16:11:31
0/0/0/0	0/0/0/0				
10.21.7.2	79	8560	8465	0	0 2d 16:10:58
0/0/0/0	0/0/0/0				

## Example: Configuring External BGP on Logical Systems with IPv6 Interfaces

This example shows how to configure external BGP (EBGP) point-to-point peer sessions on logical systems with IPv6 interfaces.

- [Requirements on page 41](#)
- [Overview on page 41](#)
- [Configuration on page 42](#)
- [Verification on page 51](#)

### Requirements

In this example, no special configuration beyond device initialization is required.

### Overview

Junos OS supports EBGP peer sessions by means of IPv6 addresses. An IPv6 peer session can be configured when an IPv6 address is specified in the **neighbor** statement. This example uses EUI-64 to generate IPv6 addresses that are automatically applied to the interfaces. An EUI-64 address is an IPv6 address that uses the IEEE EUI-64 format for the interface identifier portion of the address (the last 64 bits).



**NOTE:** Alternatively, you can configure EBGP sessions using manually assigned 128-bit IPv6 addresses.

If you use 128-bit link-local addresses for the interfaces, you must include the **local-interface** statement. This statement is valid only for 128-bit IPv6 link-local addresses and is mandatory for configuring an IPv6 EBGP link-local peer session.

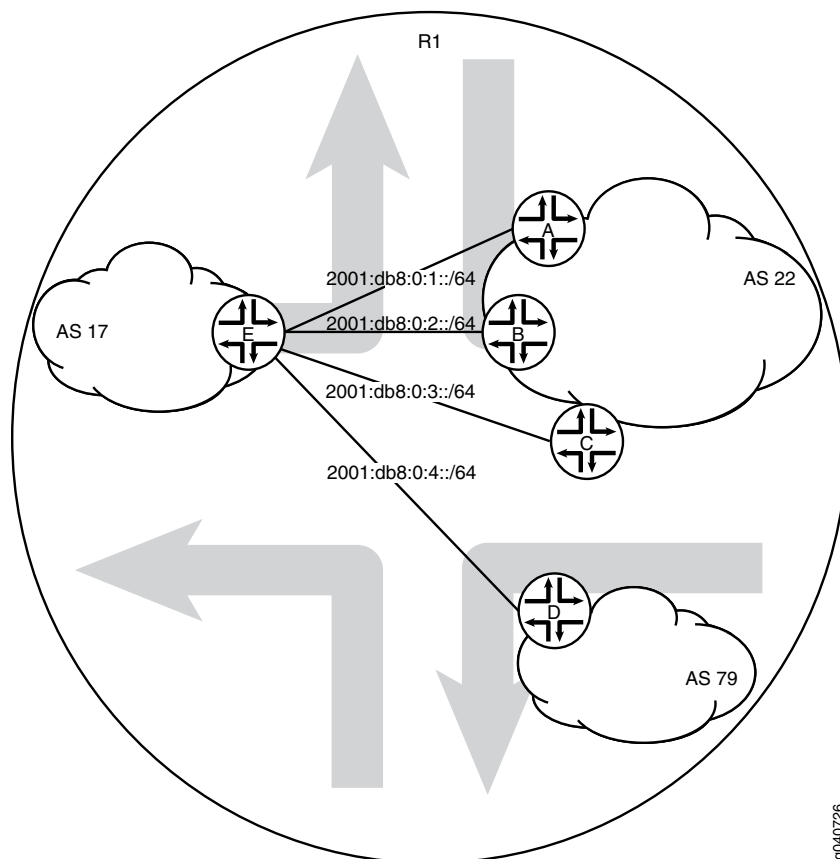
Configuring EBGP peering using link-local addresses is only applicable for directly connected interfaces. There is no support for multihop peering.

After your interfaces are up, you can use the **show interfaces terse** command to view the EUI-64-generated IPv6 addresses on the interfaces. You must use these generated addresses in the BGP **neighbor** statements. This example demonstrates the full end-to-end procedure.

In this example, Frame Relay interface encapsulation is applied to the logical tunnel (**lt**) interfaces. This is a requirement because only Frame Relay encapsulation is supported when IPv6 addresses are configured on the **lt** interfaces.

Figure 11 on page 42 shows a network with BGP peer sessions. In the sample network, Router R1 has five logical systems configured. Device E in autonomous system (AS) 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22. This example shows the step-by-step configuration on Logical System A and Logical System E.

Figure 11: Typical Network with BGP Peer Sessions



### 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 A

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-E
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation frame-relay
set logical-systems A interfaces lt-0/1/0 unit 1 dlci 1
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 25
set logical-systems A interfaces lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64
  eui-64
set logical-systems A interfaces lo0 unit 1 family inet6 address 2001:db8::1/128
set logical-systems A protocols bgp group external-peers type external
set logical-systems A protocols bgp group external-peers peer-as 17
```

```

set logical-systems A protocols bgp group external-peers neighbor
  2001:db8:0:1:2a0:a502:0:19da
set logical-systems A routing-options router-id 1.1.1.1
set logical-systems A routing-options autonomous-system 22

Device B
set logical-systems B interfaces lt-0/1/0 unit 6 description to-E
set logical-systems B interfaces lt-0/1/0 unit 6 encapsulation frame-relay
set logical-systems B interfaces lt-0/1/0 unit 6 dlci 6
set logical-systems B interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems B interfaces lt-0/1/0 unit 6 family inet6 address 2001:db8:0:2::/64
  eui-64
set logical-systems B interfaces lo0 unit 2 family inet6 address 2001:db8::2/128
set logical-systems B protocols bgp group external-peers type external
set logical-systems B protocols bgp group external-peers peer-as 17
set logical-systems B protocols bgp group external-peers neighbor
  2001:db8:0:2:2a0:a502:0:5da
set logical-systems B routing-options router-id 2.2.2.2
set logical-systems B routing-options autonomous-system 22

Device C
set logical-systems C interfaces lt-0/1/0 unit 10 description to-E
set logical-systems C interfaces lt-0/1/0 unit 10 encapsulation frame-relay
set logical-systems C interfaces lt-0/1/0 unit 10 dlci 10
set logical-systems C interfaces lt-0/1/0 unit 10 peer-unit 9
set logical-systems C interfaces lt-0/1/0 unit 10 family inet6 address 2001:db8:0:3::/64
  eui-64
set logical-systems C interfaces lo0 unit 3 family inet6 address 2001:db8::3/128
set logical-systems C protocols bgp group external-peers type external
set logical-systems C protocols bgp group external-peers peer-as 17
set logical-systems C protocols bgp group external-peers neighbor
  2001:db8:0:3:2a0:a502:0:9da
set logical-systems C routing-options router-id 3.3.3.3
set logical-systems C routing-options autonomous-system 22

Device D
set logical-systems D interfaces lt-0/1/0 unit 7 description to-E
set logical-systems D interfaces lt-0/1/0 unit 7 encapsulation frame-relay
set logical-systems D interfaces lt-0/1/0 unit 7 dlci 7
set logical-systems D interfaces lt-0/1/0 unit 7 peer-unit 21
set logical-systems D interfaces lt-0/1/0 unit 7 family inet6 address 2001:db8:0:4::/64
  eui-64
set logical-systems D interfaces lo0 unit 4 family inet6 address 2001:db8::4/128
set logical-systems D protocols bgp group external-peers type external
set logical-systems D protocols bgp group external-peers peer-as 17
set logical-systems D protocols bgp group external-peers neighbor
  2001:db8:0:4:2a0:a502:0:15da
set logical-systems D routing-options router-id 4.4.4.4
set logical-systems D routing-options autonomous-system 79

Device E
set logical-systems E interfaces lt-0/1/0 unit 5 description to-B
set logical-systems E interfaces lt-0/1/0 unit 5 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 5 dlci 6
set logical-systems E interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems E interfaces lt-0/1/0 unit 5 family inet6 address 2001:db8:0:2::/64
  eui-64
set logical-systems E interfaces lt-0/1/0 unit 9 description to-C
set logical-systems E interfaces lt-0/1/0 unit 9 encapsulation frame-relay

```

```

set logical-systems E interfaces lt-0/1/0 unit 9 dlci 10
set logical-systems E interfaces lt-0/1/0 unit 9 peer-unit 10
set logical-systems E interfaces lt-0/1/0 unit 9 family inet6 address 2001:db8:0:3::/64
eui-64
set logical-systems E interfaces lt-0/1/0 unit 21 description to-D
set logical-systems E interfaces lt-0/1/0 unit 21 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 21 dlci 7
set logical-systems E interfaces lt-0/1/0 unit 21 peer-unit 7
set logical-systems E interfaces lt-0/1/0 unit 21 family inet6 address 2001:db8:0:4::/64
eui-64
set logical-systems E interfaces lt-0/1/0 unit 25 description to-A
set logical-systems E interfaces lt-0/1/0 unit 25 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 25 dlci 1
set logical-systems E interfaces lt-0/1/0 unit 25 peer-unit 1
set logical-systems E interfaces lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64
eui-64
set logical-systems E interfaces lo0 unit 5 family inet6 address 2001:db8::5/128
set logical-systems E protocols bgp group external-peers type external
set logical-systems E protocols bgp group external-peers peer-as 22
set logical-systems E protocols bgp group external-peers neighbor
2001:db8:0:1:2a0:a502:0:1da
set logical-systems E protocols bgp group external-peers neighbor
2001:db8:0:2:2a0:a502:0:6da
set logical-systems E protocols bgp group external-peers neighbor
2001:db8:0:3:2a0:a502:0:ada
set logical-systems E protocols bgp group external-peers neighbor
2001:db8:0:4:2a0:a502:0:7da peer-as 79
set logical-systems E routing-options router-id 5.5.5.5
set logical-systems E routing-options autonomous-system 17

```

#### 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 BGP peer sessions:

1. Run the **show interfaces terse** command to verify that the physical router has a logical tunnel (lt) interface.
2. On Logical System A, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System E.

```

user@R1> show interfaces terse
Interface           Admin Link Proto  Local           Remote
...
lt-0/1/0             up    up
...

```

```

user@R1> set cli logical-system A
Logical system: A
[edit]
user@R1:A> edit
Entering configuration mode
[edit]
user@R1:A# edit interfaces
[edit interfaces]
user@R1:A# set lt-0/1/0 unit 1 encapsulation frame-relay

```

```
user@R1:A# set lt-0/1/0 unit 1 dlci 1
user@R1:A# set lt-0/1/0 unit 1 peer-unit 25
```

3. On Logical System A, configure the network address for the link to Peer E, and configure a loopback interface.

```
[edit interfaces]
user@R1:A# set lt-0/1/0 unit 1 description to-E
user@R1:A# set lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:A# set lo0 unit 1 family inet6 address 2001:db8::1/128
```

4. On Logical System E, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System A.

```
user@R1> set cli logical-system E
Logical system: E
[edit]
user@R1:E> edit
Entering configuration mode
[edit]
user@R1:E# edit interfaces
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 encapsulation frame-relay
user@R1:E# set lt-0/1/0 unit 25 dlci 1
user@R1:E# set lt-0/1/0 unit 25 peer-unit 1
```

5. On Logical System E, configure the network address for the link to Peer A, and configure a loopback interface.

```
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 description to-A
user@R1:E# set lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:E# set lo0 unit 5 family inet6 address 2001:db8::5/128
```

6. Run the **show interfaces terse** command to see the IPv6 addresses that are generated by EUI-64.

The 2001 addresses are used in this example in the BGP **neighbor** statements.



**NOTE:** The fe80 addresses are link-local addresses and are not used in this example.

```
user@R1:A> show interfaces terse
Interface          Admin Link Proto  Local                               Remote
Logical system: A

betsy@tp8:A> show interfaces terse
Interface          Admin Link Proto  Local                               Remote
lt-0/1/0
lt-0/1/0.1          up    up    inet6  2001:db8:0:1:2a0:a502:0:1da/64
                                     fe80::2a0:a502:0:1da/64
lo0
lo0.1               up    up    inet6  2001:db8::1
                                     fe80::2a0:a50f:fc56:1da

user@R1:E> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
1t-0/1/0					
1t-0/1/0.25	up	up	inet6	2001:db8:0:1:2a0:a502:0:19da/64	
				fe80::2a0:a502:0:19da/64	
1o0					
1o0.5	up	up	inet6	2001:db8::5	
				fe80::2a0:a50f:fc56:1da	

- Repeat the interface configuration on the other logical systems.

### Configuring the External BGP Sessions

**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 BGP peer sessions:

- On Logical System A, create the BGP group, and add the external neighbor address.  

```
[edit protocols bgp group external-peers]
user@R1:A# set neighbor 2001:db8:0:1:2a0:a502:0:19da
```
- On Logical System E, create the BGP group, and add the external neighbor address.  

```
[edit protocols bgp group external-peers]
user@R1:E# set neighbor 2001:db8:0:1:2a0:a502:0:1da
```
- On Logical System A, specify the autonomous system (AS) number of the external AS.  

```
[edit protocols bgp group external-peers]
user@R1:A# set peer-as 17
```
- On Logical System E, specify the autonomous system (AS) number of the external AS.  

```
[edit protocols bgp group external-peers]
user@R1:E# set peer-as 22
```
- On Logical System A, set the peer type to EBGp.  

```
[edit protocols bgp group external-peers]
user@R1:A# set type external
```
- On Logical System E, set the peer type to EBGp.  

```
[edit protocols bgp group external-peers]
user@R1:E# set type external
```
- On Logical System A, set the autonomous system (AS) number and router ID.  

```
[edit routing-options]
user@R1:A# set router-id 1.1.1.1
user@R1:A# set autonomous-system 22
```
- On Logical System E, set the AS number and router ID.  

```
[edit routing-options]
user@R1:E# set router-id 5.5.5.5
```

```
user@R1:E# set autonomous-system 17
```

9. Repeat these steps for Peers A, B, C, and D.

**Results** From configuration mode, confirm your configuration by entering the **show logical-systems** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@R1# show logical-systems
A {
  interfaces {
    lt-0/1/0 {
      unit 1 {
        description to-E;
        encapsulation frame-relay;
        dlci 1;
        peer-unit 25;
        family inet6 {
          address 2001:db8:0:1::/64 {
            eui-64;
          }
        }
      }
    }
  }
  lo0 {
    unit 1 {
      family inet6 {
        address 2001:db8::1/128;
      }
    }
  }
  protocols {
    bgp {
      group external-peers {
        type external;
        peer-as 17;
        neighbor 2001:db8:0:1:2a0:a502:0:19da;
      }
    }
    routing-options {
      router-id 1.1.1.1;
      autonomous-system 22;
    }
  }
}
B {
  interfaces {
    lt-0/1/0 {
      unit 6 {
        description to-E;
        encapsulation frame-relay;
        dlci 6;
        peer-unit 5;
        family inet6 {
```

```
        address 2001:db8:0:2::/64 {
            eui-64;
        }
    }
}
lo0 {
    unit 2 {
        family inet6 {
            address 2001:db8::2/128;
        }
    }
}
protocols {
    bgp {
        group external-peers {
            type external;
            peer-as 17;
            neighbor 2001:db8:0:2:2a0:a502:0:5da;
        }
    }
    routing-options {
        router-id 2.2.2.2;
        autonomous-system 22;
    }
}
C {
    interfaces {
        lt-0/1/0 {
            unit 10 {
                description to-E;
                encapsulation frame-relay;
                dlci 10;
                peer-unit 9;
                family inet6 {
                    address 2001:db8:0:3::/64 {
                        eui-64;
                    }
                }
            }
        }
    }
    lo0 {
        unit 3 {
            family inet6 {
                address 2001:db8::3/128;
            }
        }
    }
}
protocols {
    bgp {
        group external-peers {
            type external;
            peer-as 17;
            neighbor 2001:db8:0:3:2a0:a502:0:9da;
```

```

    }
  }
}
routing-options {
  router-id 3.3.3.3;
  autonomous-system 22;
}
}
D {
  interfaces {
    lt-0/1/0 {
      unit 7 {
        description to-E;
        encapsulation frame-relay;
        dlci 7;
        peer-unit 21;
        family inet6 {
          address 2001:db8:0:4::/64 {
            eui-64;
          }
        }
      }
    }
  }
  lo0 {
    unit 4 {
      family inet6 {
        address 2001:db8::4/128;
      }
    }
  }
}
protocols {
  bgp {
    group external-peers {
      type external;
      peer-as 17;
      neighbor 2001:db8:0:4:2a0:a502:0:15da;
    }
  }
  routing-options {
    router-id 4.4.4.4;
    autonomous-system 79;
  }
}
E {
  interfaces {
    lt-0/1/0 {
      unit 5 {
        description to-B;
        encapsulation frame-relay;
        dlci 6;
        peer-unit 6;
        family inet6 {
          address 2001:db8:0:2::/64 {
            eui-64;
          }
        }
      }
    }
  }
}

```

```
    }
  }
  unit 9 {
    description to-C;
    encapsulation frame-relay;
    dlci 10;
    peer-unit 10;
    family inet6 {
      address 2001:db8:0:3::/64 {
        eui-64;
      }
    }
  }
  unit 21 {
    description to-D;
    encapsulation frame-relay;
    dlci 7;
    peer-unit 7;
    family inet6 {
      address 2001:db8:0:4::/64 {
        eui-64;
      }
    }
  }
  unit 25 {
    description to-A;
    encapsulation frame-relay;
    dlci 1;
    peer-unit 1;
    family inet6 {
      address 2001:db8:0:1::/64 {
        eui-64;
      }
    }
  }
}
lo0 {
  unit 5 {
    family inet6 {
      address 2001:db8::5/128;
    }
  }
}
}
protocols {
  bgp {
    group external-peers {
      type external;
      peer-as 22;
      neighbor 2001:db8:0:1:2a0:a502:0:1da;
      neighbor 2001:db8:0:2:2a0:a502:0:6da;
      neighbor 2001:db8:0:3:2a0:a502:0:ada;
      neighbor 2001:db8:0:4:2a0:a502:0:7da {
        peer-as 79;
      }
    }
  }
}
```

```

    }
  }
  routing-options {
    router-id 5.5.5.5;
    autonomous-system 17;
  }
}

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 51](#)
- [Verifying BGP Groups on page 54](#)
- [Verifying BGP Summary Information on page 54](#)
- [Checking the Routing Table on page 54](#)

#### *Verifying BGP Neighbors*

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

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

```

user@R1:E> show bgp neighbor
Peer: 2001:db8:0:1:2a0:a502:0:1da+54987 AS 22 Local:
2001:db8:0:1:2a0:a502:0:19da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 20 Recv: 0
  Peer ID: 1.1.1.1      Local ID: 5.5.5.5      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 0
  BFD: disabled, down
  Local Interface: lt-0/1/0.25
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        0
    Accepted prefixes:        0

```

```

    Suppressed due to damping:    0
    Advertised prefixes:          0
    Last traffic (seconds): Received 7    Sent 18    Checked 81
    Input messages:  Total 1611    Updates 1      Refreshes 0      Octets 30660
    Output messages: Total 1594    Updates 0      Refreshes 0      Octets 30356
    Output Queue[0]: 0

Peer: 2001:db8:0:2:2a0:a502:0:6da+179 AS 22 Local:
2001:db8:0:2:2a0:a502:0:5da+55502 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 26 Recv: 0
  Peer ID: 2.2.2.2      Local ID: 5.5.5.5      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 2
  BFD: disabled, down
  Local Interface: lt-0/1/0.5
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        0
    Accepted prefixes:        0
    Suppressed due to damping: 0
    Advertised prefixes:      0
    Last traffic (seconds): Received 15    Sent 8      Checked 8
    Input messages:  Total 1610    Updates 1      Refreshes 0      Octets 30601
    Output messages: Total 1645    Updates 0      Refreshes 0      Octets 32417
    Output Queue[0]: 0

Peer: 2001:db8:0:3:2a0:a502:0:ada+55983 AS 22 Local:
2001:db8:0:3:2a0:a502:0:9da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 3.3.3.3      Local ID: 5.5.5.5      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 3
  BFD: disabled, down
  Local Interface: lt-0/1/0.9
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300

```

```

Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet6-unicast
NLRI of received end-of-rib markers: inet6-unicast
NLRI of all end-of-rib markers sent: inet6-unicast
Peer supports 4 byte AS extension (peer-as 22)
Peer does not support Addpath
Table inet6.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      0
Last traffic (seconds): Received 21   Sent 21   Checked 67
Input messages:  Total 1610   Updates 1       Refreshes 0       Octets 30641
Output messages: Total 1587   Updates 0       Refreshes 0       Octets 30223
Output Queue[0]: 0

Peer: 2001:db8:0:4:2a0:a502:0:7da+49255 AS 79 Local:
2001:db8:0:4:2a0:a502:0:15da+179 AS 17
  Type: External   State: Established   Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 4.4.4.4      Local ID: 5.5.5.5      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 1
  BFD: disabled, down
  Local Interface: lt-0/1/0.21
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 79)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        0
    Accepted prefixes:        0
    Suppressed due to damping: 0
    Advertised prefixes:      0
  Last traffic (seconds): Received 6    Sent 17    Checked 25
  Input messages:  Total 1615   Updates 1       Refreshes 0       Octets 30736
  Output messages: Total 1593   Updates 0       Refreshes 0       Octets 30337
  Output Queue[0]: 0

```

**Meaning** IPv6 unicast network layer reachability information (NLRI) is being exchanged between the neighbors.

**Verifying BGP Groups**

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From operational mode, run the **show bgp group** command.

```
user@R1:~> show bgp group
Group Type: External                               Local AS: 17
  Name: external-peers  Index: 0                   Flags: <>
  Holdtime: 0
  Total peers: 4      Established: 4
  2001:db8:0:1:2a0:a502:0:1da+54987
  2001:db8:0:2:2a0:a502:0:6da+179
  2001:db8:0:3:2a0:a502:0:ada+55983
  2001:db8:0:4:2a0:a502:0:7da+49255
  inet6.0: 0/0/0/0

Groups: 1  Peers: 4   External: 4   Internal: 0   Down peers: 0   Flaps: 0
Table      Tot Paths  Act Paths  Suppressed  History  Damp State  Pending
inet6.0           0         0         0         0        0        0        0
inet6.2           0         0         0         0        0        0        0
```

**Meaning** The group type is external, and the group has four peers.

**Verifying BGP Summary Information**

**Purpose** Verify that the BGP that the peer relationships are established.

**Action** From operational mode, run the **show bgp summary** command.

```
user@R1:~> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table      Tot Paths  Act Paths  Suppressed  History  Damp State  Pending
inet6.0           0         0         0         0        0        0        0
inet6.2           0         0         0         0        0        0        0
Peer      AS      InPkt    OutPkt    OutQ    Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
2001:db8:0:1:2a0:a502:0:1da      22    1617    1600      0      0
12:07:00 Establ
inet6.0: 0/0/0/0
2001:db8:0:2:2a0:a502:0:6da      22    1616    1651      0      0
12:06:56 Establ
inet6.0: 0/0/0/0
2001:db8:0:3:2a0:a502:0:ada      22    1617    1594      0      0
12:04:32 Establ
inet6.0: 0/0/0/0
2001:db8:0:4:2a0:a502:0:7da      79    1621    1599      0      0
12:07:00 Establ
inet6.0: 0/0/0/0
```

**Meaning** The Down peers: 0 output shows that the BGP peers are in the established state.

**Checking the Routing Table**

**Purpose** Verify that the inet6.0 routing table is populated with local and direct routes.

**Action** From operational mode, run the **show route** command.

```

user@R1:E> show route
inet6.0: 15 destinations, 18 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::5/128    *[Direct/0] 12:41:18
                  > via lo0.5
2001:db8:0:1::/64  *[Direct/0] 14:40:01
                  > via lt-0/1/0.25
2001:db8:0:1:2a0:a502:0:19da/128
                  *[Local/0] 14:40:01
                  Local via lt-0/1/0.25
2001:db8:0:2::/64  *[Direct/0] 14:40:02
                  > via lt-0/1/0.5
2001:db8:0:2:2a0:a502:0:5da/128
                  *[Local/0] 14:40:02
                  Local via lt-0/1/0.5
2001:db8:0:3::/64  *[Direct/0] 14:40:02
                  > via lt-0/1/0.9
2001:db8:0:3:2a0:a502:0:9da/128
                  *[Local/0] 14:40:02
                  Local via lt-0/1/0.9
2001:db8:0:4::/64  *[Direct/0] 14:40:01
                  > via lt-0/1/0.21
2001:db8:0:4:2a0:a502:0:15da/128
                  *[Local/0] 14:40:01
                  Local via lt-0/1/0.21
fe80::/64         *[Direct/0] 14:40:02
                  > via lt-0/1/0.5
                  [Direct/0] 14:40:02
                  > via lt-0/1/0.9
                  [Direct/0] 14:40:01
                  > via lt-0/1/0.21
                  [Direct/0] 14:40:01
                  > via lt-0/1/0.25
fe80::2a0:a502:0:5da/128
                  *[Local/0] 14:40:02
                  Local via lt-0/1/0.5
fe80::2a0:a502:0:9da/128
                  *[Local/0] 14:40:02
                  Local via lt-0/1/0.9
fe80::2a0:a502:0:15da/128
                  *[Local/0] 14:40:01
                  Local via lt-0/1/0.21
fe80::2a0:a502:0:19da/128
                  *[Local/0] 14:40:01
                  Local via lt-0/1/0.25
fe80::2a0:a50f:fc56:1da/128
                  *[Direct/0] 12:41:18
                  > via lo0.5

```

**Meaning** The inet6.0 routing table contains local and direct routes. To populate the routing table with other types of routes, you must configure routing policies.

**Related Documentation**

- [Examples: Configuring Internal BGP Peering on page 56](#)
- [BGP Configuration Overview on page 12](#)

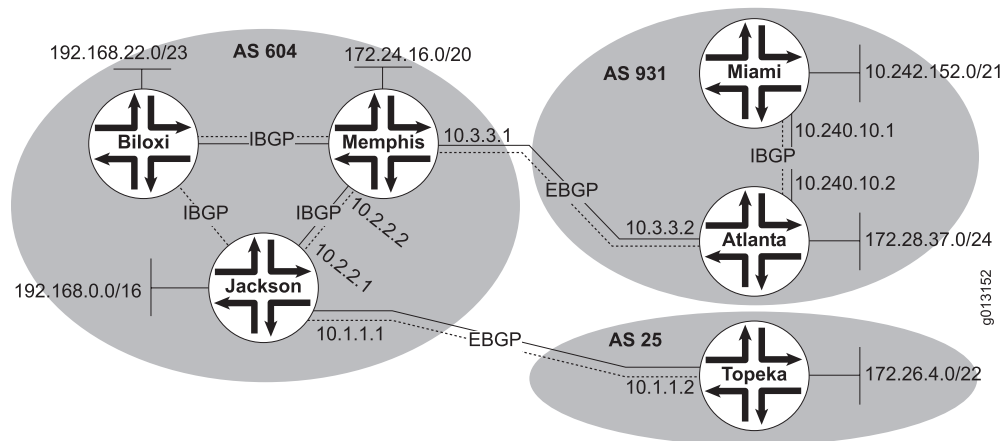
## Examples: Configuring Internal BGP Peering

- [Understanding Internal BGP Peering Sessions on page 56](#)
- [Example: Configuring Internal BGP Peer Sessions on page 57](#)
- [Example: Configuring Internal BGP Peering Sessions on Logical Systems on page 68](#)

### Understanding Internal BGP Peering Sessions

When two BGP-enabled devices are in the same autonomous system (AS), the BGP session is called an *internal* BGP session, or IBGP session. BGP uses the same message types on IBGP and external BGP (EBGP) sessions, but the rules for when to send each message and how to interpret each message differ slightly. For this reason, some people refer to IBGP and EBGP as two separate protocols.

Figure 12: Internal and External BGP



In [Figure 12 on page 56](#), Device Jackson, Device Memphis, and Device Biloxi have IBGP peer sessions with each other. Likewise, Device Miami and Device Atlanta have IBGP peer sessions between each other.

The purpose of IBGP is to provide a means by which EBGP route advertisements can be forwarded throughout the network. In theory, to accomplish this task you could redistribute all of your EBGP routes into an interior gateway protocol (IGP), such as OSPF or IS-IS. This, however, is not recommended in a production environment because of the large number of EBGP routes in the Internet and because of the way that IGPs operate. In short, with that many routes the IGP churns or crashes.

Generally, the loopback interface (lo0) is used to establish connections between IBGP peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the IBGP peering session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peering session also goes up and down. Thus the loopback interface provides fault tolerance in case the physical interface or the link goes down, if the device has link redundancy.

While IBGP neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every

other device through neighbor peer relationships. The **neighbor** statement creates the mesh. Because of the full mesh requirement of IBGP, you must configure individual peering sessions between all IBGP devices in the AS. The full mesh need not be physical links. Rather, the configuration on each routing device must create a full mesh of peer sessions (using multiple **neighbor** statements).



**NOTE:** The requirement for a full mesh is waived if you configure a confederation or route reflection.

To understand the full-mesh requirement, consider that an IBGP-learned route cannot be readvertised to another IBGP peer. The reason for preventing the readvertisement of IBGP routes and requiring the full mesh is to avoid routing loops within an AS. The AS path attribute is the means by which BGP routing devices avoid loops. The path information is examined for the local AS number only when the route is received from an EBGP peer. Because the attribute is only modified across AS boundaries, this system works well. However, the fact that the attribute is only modified across AS boundaries presents an issue inside the AS. For example, suppose that routing devices A, B, and C are all in the same AS. Device A receives a route from an EBGP peer and sends the route to Device B, which installs it as the active route. The route is then sent to Device C, which installs it locally and sends it back to Device A. If Device A installs the route, a loop is formed within the AS. The routing devices are not able to detect the loop because the AS path attribute is not modified during these advertisements. Therefore, the BGP protocol designers decided that the only assurance of never forming a routing loop was to prevent an IBGP peer from advertising an IBGP-learned route within the AS. For route reachability, the IBGP peers are fully meshed.

IBGP supports multihop connections, so IBGP neighbors can be located anywhere within the AS and often do not share a link. A recursive route lookup resolves the loopback peering address to an IP forwarding next hop. The lookup service is provided by static routes or an IGP such as OSPF, or BGP routes.

## Example: Configuring Internal BGP Peer Sessions

This example shows how to configure internal BGP peer sessions.

- [Requirements on page 57](#)
- [Overview on page 57](#)
- [Configuration on page 59](#)
- [Verification on page 66](#)

### Requirements

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

### Overview

In this example, you configure internal BGP (IBGP) peer sessions. The loopback interface (lo0) is used to establish connections between IBGP peers. The loopback interface is

always up as long as the device is operating. If there is a route to the loopback address, the IBGP peer session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peer session also goes up and down. Thus, if the device has link redundancy, the loopback interface provides fault tolerance in case the physical interface or one of the links goes down.

When a device peers with a remote device's loopback interface address, the local device expects BGP update messages to come from (be sourced by) the remote device's loopback interface address. The **local-address** statement enables you to specify the source information in BGP update messages. If you omit the **local-address** statement, the expected source of BGP update messages is based on the device's source address selection rules, which normally results in the egress interface address being the expected source of update messages. When this happens, the peer session is not established because a mismatch exists between the expected source address (the egress interface of the peer) and the actual source (the loopback interface of the peer). To make sure that the expected source address matches the actual source address, specify the loopback interface address in the **local-address** statement.

Because IBGP supports multihop connections, IBGP neighbors can be located anywhere within the autonomous system (AS) and often do not share a link. A recursive route lookup resolves the loopback peer address to an IP forwarding next hop. In this example, this service is provided by OSPF. Although interior gateway protocol (IGP) neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every other device through neighbor peer relationships. The **neighbor** statement creates the mesh.



**NOTE:** The requirement for a full mesh is waived if you configure a confederation or route reflection.

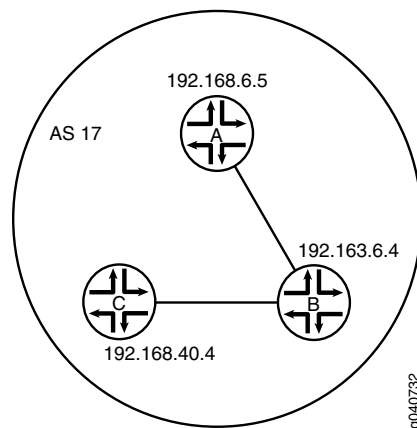
---

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP routing information base (RIB) and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

[Figure 13 on page 59](#) shows a typical network with internal peer sessions.

Figure 13: Typical Network with IBGP Sessions



### Configuration

- [Configuring Device A on page 60](#)
- [Configuring Device B on page 62](#)
- [Configuring Device C on page 64](#)

#### 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 A

```
set interfaces ge-0/1/0 unit 1 description to-B
set interfaces ge-0/1/0 unit 1 family inet address 10.10.10.1/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description "connections to B and C"
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.1
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

##### Device B

```
set interfaces ge-0/1/0 unit 2 description to-A
set interfaces ge-0/1/0 unit 2 family inet address 10.10.10.2/30
set interfaces ge-0/1/1 unit 5 description to-C
set interfaces ge-0/1/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description "connections to A and C"
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.6.5
```

```

set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.2
set protocols ospf area 0.0.0.0 interface ge-0/1/1.5
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17

```

**Device C**

```

set interfaces ge-0/1/0 unit 6 description to-B
set interfaces ge-0/1/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description "connections to A and B"
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.6
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17

```

### Configuring Device A

**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 internal BGP peer sessions on Device A:

1. Configure the interfaces.

```

[edit interfaces ge-0/1/0 unit 1]
user@A# set description to-B
user@A# set family inet address 10.10.10.1/30

```

```

[edit interfaces]
user@A# set lo0 unit 1 family inet address 192.168.6.5/32

```

2. Configure BGP.

The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```

[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set description "connections to B and C"
user@A# set local-address 192.168.6.5
user@A# set export send-direct
user@A# set neighbor 192.163.6.4
user@A# set neighbor 192.168.40.4

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]

```

```

user@A# set interface lo0.1 passive
user@A# set interface ge-0/1/0.1

```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 2]
user@A# set from protocol direct
user@A# set then accept

```

5. Configure the router ID and the AS number.

```

[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17

```

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

```

user@A# show interfaces
ge-0/1/0 {
  unit 1 {
    description to-B;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}

user@A# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}

user@A# show protocols
bgp {
  group internal-peers {
    type internal;
    description "connections to B and C";
    local-address 192.168.6.5;
    export send-direct;
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
  }
}

```

```

    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.1 {
        passive;
      }
      interface ge-0/1/0.1;
    }
  }
}

user@A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;

```

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

### Configuring Device B

**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*.

To configure internal BGP peer sessions on Device B:

1. Configure the interfaces.

```

[edit interfaces ge-0/1/0 unit 2]
user@B# set description to-A
user@B# set family inet address 10.10.10.2/30

```

```

[edit interfaces ge-0/1/1]
user@B# set unit 5 description to-C
user@B# set unit 5 family inet address 10.10.10.5/30

```

```

[edit interfaces]
user@B# set lo0 unit 2 family inet address 192.163.6.4/32

```

2. Configure BGP.

The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```

[edit protocols bgp group internal-peers]
user@B# set type internal
user@B# set description "connections to A and C"
user@B# set local-address 192.163.6.4
user@B# set export send-direct
user@B# set neighbor 192.168.40.4
user@B# set neighbor 192.168.6.5

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@B# set interface lo0.2 passive
user@B# set interface ge-0/1/0.2
user@B# set interface ge-0/1/1.5

```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@B# set from protocol direct
user@B# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@B# set router-id 192.163.6.4
user@B# set autonomous-system 17
```

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

```
user@B# show interfaces
ge-0/1/0 {
  unit 2 {
    description to-A;
    family inet {
      address 10.10.10.2/30;
    }
  }
}
ge-0/1/1 {
  unit 5 {
    description to-C;
    family inet {
      address 10.10.10.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.163.6.4/32;
    }
  }
}

user@B# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}

user@B# show protocols
bgp {
  group internal-peers {
    type internal;
```

```

        description "connections to A and C";
        local-address 192.163.6.4;
        export send-direct;
        neighbor 192.168.40.4;
        neighbor 192.168.6.5;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.2 {
            passive;
        }
        interface ge-0/1/0.2;
        interface ge-0/1/1.5;
    }
}

```

```

user@B# show routing-options
router-id 192.163.6.4;
autonomous-system 17;

```

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

### Configuring Device C

**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 internal BGP peer sessions on Device C:

1. Configure the interfaces.

```

[edit interfaces ge-0/1/0 unit 6]
user@C# set description to-B
user@C# set family inet address 10.10.10.6/30

[edit interfaces]
user@C# set lo0 unit 3 family inet address 192.168.40.4/32

```

2. Configure BGP.

The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```

[edit protocols bgp group internal-peers]
user@C# set type internal
user@C# set description "connections to A and B"
user@C# set local-address 192.168.40.4
user@C# set export send-direct
user@C# set neighbor 192.163.6.4
user@C# set neighbor 192.168.6.5

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@C# set interface lo0.3 passive
user@C# set interface ge-0/1/0.6

```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@C# set from protocol direct
user@C# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@C# set router-id 192.168.40.4
user@C# set autonomous-system 17
```

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

```
user@C# show interfaces
ge-0/1/0 {
  unit 6 {
    description to-B;
    family inet {
      address 10.10.10.6/30;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.40.4/32;
    }
  }
}

user@C# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}

user@C# show protocols
bgp {
  group internal-peers {
    type internal;
    description "connections to A and B";
    local-address 192.168.40.4;
    export send-direct;
    neighbor 192.163.6.4;
    neighbor 192.168.6.5;
  }
}
ospf {
```

```

area 0.0.0.0 {
  interface lo0.3 {
    passive;
  }
  interface ge-0/1/0.6;
}
}

user@C# show routing-options
router-id 192.168.40.4;
autonomous-system 17;

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 66](#)
- [Verifying BGP Groups on page 67](#)
- [Verifying BGP Summary Information on page 68](#)
- [Verifying That BGP Routes Are Installed in the Routing Table on page 68](#)

#### *Verifying BGP Neighbors*

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

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

```

user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: Sync
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: Preference LocalAddress Refresh
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4    Local ID: 192.168.6.5    Active Holdtime: 90
  Keepalive Interval: 30    Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete

```

```

Send state: in sync
Active prefixes:          0
Received prefixes:        3
Accepted prefixes:        3
Suppressed due to damping: 0
Advertised prefixes:      2
Last traffic (seconds): Received 25   Sent 19   Checked 67
Input messages:  Total 2420   Updates 4     Refreshes 0     Octets 46055
Output messages: Total 2411   Updates 2     Refreshes 0     Octets 45921
Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17   Local: 192.168.6.5+56466 AS 17
Type: Internal   State: Established   Flags: Sync
Last State: OpenConfirm   Last Event: RecvKeepAlive
Last Error: None
Export: [ send-direct ]
Options: Preference LocalAddress Refresh
Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 192.168.40.4   Local ID: 192.168.6.5   Active Holdtime: 90
Keepalive Interval: 30   Peer index: 1
BFD: disabled, down
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 17)
Peer does not support Addpath
Table inet.0 Bit: 10000
RIB State: BGP restart is complete
Send state: in sync
Active prefixes:          0
Received prefixes:        2
Accepted prefixes:        2
Suppressed due to damping: 0
Advertised prefixes:      2
Last traffic (seconds): Received 7     Sent 21   Checked 24
Input messages:  Total 2412   Updates 2     Refreshes 0     Octets 45867
Output messages: Total 2409   Updates 2     Refreshes 0     Octets 45883
Output Queue[0]: 0

```

### Verifying BGP Groups

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From operational mode, enter the **show bgp group** command.

```

user@A> show bgp group
Group Type: Internal   AS: 17                               Local AS: 17
Name: internal-peers  Index: 0                               Flags: <Export Eval>
Export: [ send-direct ]
Holdtime: 0
Total peers: 2         Established: 2
192.163.6.4+179

```

```

192.168.40.4+179
inet.0: 0/5/5/0

Groups: 1 Peers: 2 External: 0 Internal: 2 Down peers: 0 Flaps: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 5 0 0 0 0 0 0

```

### Verifying BGP Summary Information

**Purpose** Verify that the BGP configuration is correct.

**Action** From operational mode, enter the **show bgp summary** command.

```

user@A> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 5 0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4 17 2441 2432 0 0 18:18:52
0/3/3/0 0/0/0/0
192.168.40.4 17 2432 2430 0 0 18:18:48
0/2/2/0 0/0/0/0

```

### Verifying That BGP Routes Are Installed in the Routing Table

**Purpose** Verify that the export policy configuration is causing the BGP routes to be installed in the routing tables of the peers.

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

```

user@A> show route protocol bgp
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30 [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
AS path: I
> to 10.10.10.2 via ge-0/1/0.1
10.10.10.4/30 [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
AS path: I
> to 10.10.10.2 via ge-0/1/0.1
[BGP/170] 07:07:12, localpref 100, from 192.168.40.4
AS path: I
> to 10.10.10.2 via ge-0/1/0.1
192.163.6.4/32 [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
AS path: I
> to 10.10.10.2 via ge-0/1/0.1
192.168.40.4/32 [BGP/170] 07:07:12, localpref 100, from 192.168.40.4
AS path: I
> to 10.10.10.2 via ge-0/1/0.1

```

## Example: Configuring Internal BGP Peering Sessions on Logical Systems

This example shows how to configure internal BGP peer sessions on logical systems.

- [Requirements on page 69](#)
- [Overview on page 69](#)

- [Configuration on page 69](#)
- [Verification on page 76](#)

## Requirements

In this example, no special configuration beyond device initialization is required.

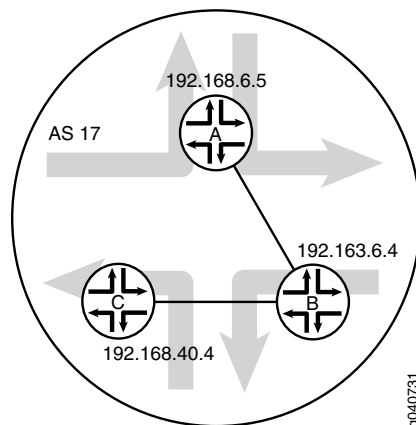
## Overview

In this example, you configure internal BGP (IBGP) peering sessions.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

[Figure 14 on page 69](#) shows a typical network with internal peer sessions.

**Figure 14: Typical Network with IBGP Sessions**



## 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.

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-B
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 2
set logical-systems A interfaces lt-0/1/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-0/1/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
```

```

set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
set logical-systems B interfaces lt-0/1/0 unit 2 description to-A
set logical-systems B interfaces lt-0/1/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-0/1/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-0/1/0 unit 5 description to-C
set logical-systems B interfaces lt-0/1/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-0/1/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17
set logical-systems C interfaces lt-0/1/0 unit 6 description to-B
set logical-systems C interfaces lt-0/1/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-0/1/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-0/1/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17

```

### Device A

**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 internal BGP peer sessions on Device A:

1. Configure the interfaces.
 

```

[edit logical-systems A interfaces lt-0/1/0 unit 1]
user@R1# set description to-B
user@R1# set encapsulation ethernet
user@R1# set peer-unit 2

```

```

user@R1# set family inet address 10.10.10.1/30
user@R1# set family inet address 192.168.6.5/32
user@R1# up
user@R1# up
[edit logical-systems A interfaces]
user@R1# set lo0 unit 1 family inet address 192.168.6.5/32
user@R1# exit
[edit]
user@R1# edit logical-systems B interfaces lt-0/1/0
[edit logical-systems B interfaces lt-0/1/0]
user@R1# set unit 2 description to-A
user@R1# set unit 2 encapsulation ethernet
user@R1# set unit 2 peer-unit 1
user@R1# set unit 2 family inet address 10.10.10.2/30
user@R1# set unit 5 description to-C
user@R1# set unit 5 encapsulation ethernet
user@R1# set unit 5 peer-unit 6
user@R1# set family inet address 10.10.10.5/30
user@R1# up
[edit logical-systems B interfaces]
user@R1# set lo0 unit 2 family inet address 192.163.6.4/32
user@R1# exit
[edit]
user@R1# edit logical-systems C interfaces lt-0/1/0 unit 6
[edit logical-systems C interfaces lt-0/1/0 unit 6]
set description to-B
set encapsulation ethernet
set peer-unit 5
set family inet address 10.10.10.6/30
user@R1# up
user@R1# up
[edit logical-systems C interfaces]
set lo0 unit 3 family inet address 192.168.40.4/32

```

## 2. Configure BGP.

On Logical System A, the **neighbor** statements are included for both Device B and Device C, even though Logical System A is not directly connected to Device C.

```

[edit logical-systems A protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.6.5
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.40.4

```

```

[edit logical-systems B protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.163.6.4
user@R1# set export send-direct
user@R1# set neighbor 192.168.40.4
user@R1# set neighbor 192.168.6.5

```

```

[edit logical-systems C protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.40.4

```

```
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface lt-0/1/0.1
```

```
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.2 passive
user@R1# set interface lt-0/1/0.2
user@R1# set interface lt-0/1/0.5
```

```
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.3 passive
user@R1# set interface lt-0/1/0.6
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit logical-systems A policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
```

```
[edit logical-systems B policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
```

```
[edit logical-systems C policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and the autonomous system (AS) number.

```
[edit logical-systems A routing-options]
user@R1# set router-id 192.168.6.5
user@R1# set autonomous-system 17
```

```
[edit logical-systems B routing-options]
user@R1# set router-id 192.163.6.4
user@R1# set autonomous-system 17
```

```
[edit logical-systems C routing-options]
user@R1# set router-id 192.168.40.4
user@R1# set autonomous-system 17
```

**Results** From configuration mode, confirm your configuration by entering the **show logical-systems** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show logical-systems
```

```

A {
  interfaces {
    lt-0/1/0 {
      unit 1 {
        description to-B;
        encapsulation ethernet;
        peer-unit 2;
        family inet {
          address 10.10.10.1/30;
        }
      }
    }
    lo0 {
      unit 1 {
        family inet {
          address 192.168.6.5/32;
        }
      }
    }
  }
  protocols {
    bgp {
      group internal-peers {
        type internal;
        local-address 192.168.6.5;
        export send-direct;
        neighbor 192.163.6.4;
        neighbor 192.168.40.4;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.1 {
          passive;
        }
        interface lt-0/1/0.1;
      }
    }
  }
  policy-options {
    policy-statement send-direct {
      term 2 {
        from protocol direct;
        then accept;
      }
    }
  }
  routing-options {
    router-id 192.168.6.5;
    autonomous-system 17;
  }
}
B {
  interfaces {
    lt-0/1/0 {
      unit 2 {

```

```
        description to-A;
        encapsulation ethernet;
        peer-unit 1;
        family inet {
            address 10.10.10.2/30;
        }
    }
    unit 5 {
        description to-C;
        encapsulation ethernet;
        peer-unit 6;
        family inet {
            address 10.10.10.5/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.163.6.4/32;
        }
    }
}
}
protocols {
    bgp {
        group internal-peers {
            type internal;
            local-address 192.163.6.4;
            export send-direct;
            neighbor 192.168.40.4;
            neighbor 192.168.6.5;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.2 {
                passive;
            }
            interface lt-0/1/0.2;
            interface lt-0/1/0.5;
        }
    }
}
policy-options {
    policy-statement send-direct {
        term 2 {
            from protocol direct;
            then accept;
        }
    }
}
}
routing-options {
    router-id 192.163.6.4;
    autonomous-system 17;
}
```

```

}
C {
  interfaces {
    lt-0/1/0 {
      unit 6 {
        description to-B;
        encapsulation ethernet;
        peer-unit 5;
        family inet {
          address 10.10.10.6/30;
        }
      }
    }
    lo0 {
      unit 3 {
        family inet {
          address 192.168.40.4/32;
        }
      }
    }
  }
  protocols {
    bgp {
      group internal-peers {
        type internal;
        local-address 192.168.40.4;
        export send-direct;
        neighbor 192.163.6.4;
        neighbor 192.168.6.5;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.3 {
          passive;
        }
        interface lt-0/1/0.6;
      }
    }
  }
  policy-options {
    policy-statement send-direct {
      term 2 {
        from protocol direct;
        then accept;
      }
    }
  }
  routing-options {
    router-id 192.168.40.4;
    autonomous-system 17;
  }
}

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 76](#)
- [Verifying BGP Groups on page 77](#)
- [Verifying BGP Summary Information on page 77](#)
- [Verifying That BGP Routes Are Installed in the Routing Table on page 78](#)

### Verifying BGP Neighbors

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

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

```

user@R1> show bgp neighbor logical-system A
Peer: 192.163.6.4+179 AS 17      Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4      Local ID: 192.168.6.5      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        3
    Accepted prefixes:        3
    Suppressed due to damping: 0
    Advertised prefixes:      2
  Last traffic (seconds): Received 16    Sent 1    Checked 63
  Input messages:  Total 15713  Updates 4    Refreshes 0    Octets 298622
  Output messages: Total 15690  Updates 2    Refreshes 0    Octets 298222
  Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17      Local: 192.168.6.5+56466 AS 17
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None

```

```

Export: [ send-direct ]
Options: <Preference LocalAddress Refresh>
Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 192.168.40.4    Local ID: 192.168.6.5    Active Holdtime: 90
Keepalive Interval: 30    Peer index: 1
BFD: disabled, down
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 17)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:      0
  Received prefixes:    2
  Accepted prefixes:    2
  Suppressed due to damping: 0
  Advertised prefixes:  2
Last traffic (seconds): Received 15    Sent 22    Checked 68
Input messages: Total 15688 Updates 2    Refreshes 0    Octets 298111
Output messages: Total 15688 Updates 2    Refreshes 0    Octets 298184
Output Queue[0]: 0

```

### Verifying BGP Groups

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From the operational mode, enter the **show bgp group** command.

```

user@A> show bgp group logical-system A
Group Type: Internal    AS: 17                      Local AS: 17
Name: internal-peers   Index: 0                    Flags: <Export Eval>
Export: [ send-direct ]
Holdtime: 0
Total peers: 2          Established: 2
192.163.6.4+179
192.168.40.4+179
inet.0: 0/5/5/0

Groups: 1  Peers: 2   External: 0   Internal: 2   Down peers: 0   Flaps: 0
Table      Tot Paths  Act Paths  Suppressed   History Damp State   Pending
inet.0           5           0           0           0         0         0

```

### Verifying BGP Summary Information

**Purpose** Verify that the BGP configuration is correct.

**Action** From the operational mode, enter the **show bgp summary** command.

```

user@A> show bgp summary logical-system A

```

```

Groups: 1 Peers: 2 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History  Damp State  Pending
inet.0      5          0          0          0        0      0      0
Peer        AS      InPkt    OutPkt    OutQ    Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4 17      15723    15700     0        0 4d 22:13:15
0/3/3/0      0/0/0/0
192.168.40.4 17      15698    15699     0        0 4d 22:13:11
0/2/2/0      0/0/0/0

```

### Verifying That BGP Routes Are Installed in the Routing Table

**Purpose** Verify that the export policy configuration is working.

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

```

user@A> show route protocol bgp logical-system A
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                  AS path: I
                  > to 10.10.10.2 via lt-0/1/0.1
10.10.10.4/30      [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                  AS path: I
                  > to 10.10.10.2 via lt-0/1/0.1
                  [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via lt-0/1/0.1
192.163.6.4/32     [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                  AS path: I
                  > to 10.10.10.2 via lt-0/1/0.1
192.168.40.4/32    [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via lt-0/1/0.1

```

**Related Documentation**

- [Examples: Configuring External BGP Peering on page 33](#)

## CHAPTER 5

# BGP Path Attribute Configuration

- [Example: Configuring BGP Local Preference on page 79](#)
- [Examples: Configuring BGP MED on page 92](#)
- [Examples: Configuring BGP Local AS on page 131](#)
- [Example: Configuring the Accumulated IGP Attribute for BGP on page 151](#)
- [Example: Configuring AS Override on page 190](#)
- [Example: Disabling Suppression of Route Advertisements on page 200](#)
- [Configuring 4-Byte Autonomous System Numbers on page 208](#)
- [Disabling Attribute Set Messages on Independent AS Domains for BGP Loop Detection on page 209](#)

### Example: Configuring BGP Local Preference

---

- [Understanding the BGP Local Preference on page 79](#)
- [Example: Configuring the Local Preference Value for BGP Routes on page 80](#)

### Understanding the BGP Local Preference

Internal BGP (IBGP) sessions use a metric called the *local preference*, which is carried in IBGP update packets in the path attribute LOCAL\_PREF. When an autonomous system (AS) has multiple routes to another AS, the local preference indicates the degree of preference for one route over the other routes. The route with the highest local preference value is preferred.

The LOCAL\_PREF path attribute is always advertised to IBGP peers and to neighboring confederations. It is never advertised to external BGP (EBGP) peers. The default behavior is to not modify the LOCAL\_PREF path attribute if it is present.

The LOCAL\_PREF path attribute applies at export time only, when the routes are exported from the routing table into BGP.

If a BGP route is received without a LOCAL\_PREF attribute, the route is stored in the routing table and advertised by BGP as if it were received with a LOCAL\_PREF value of 100. A non-BGP route that is advertised by BGP is advertised with a LOCAL\_PREF value of 100 by default.

## Example: Configuring the Local Preference Value for BGP Routes

This example shows how to configure local preference in internal BGP (IBGP) peer sessions.

- [Requirements on page 80](#)
- [Overview on page 80](#)
- [Configuration on page 81](#)
- [Verification on page 90](#)

### Requirements

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

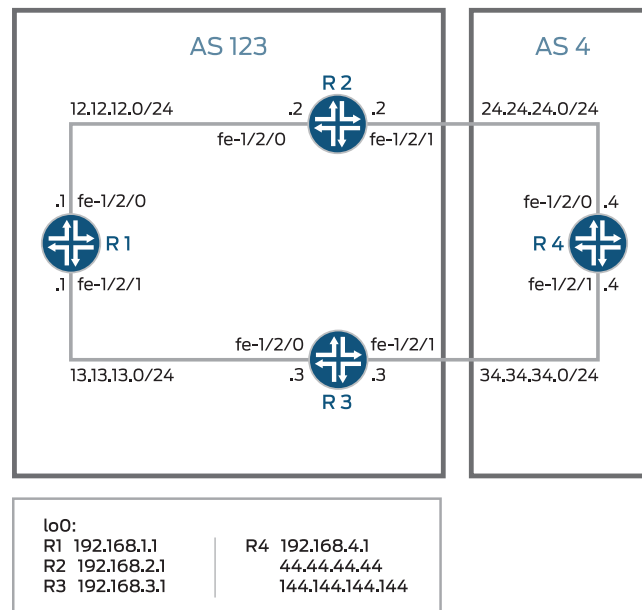
### Overview

To change the local preference metric advertised in the path attribute, you must include the **local-preference** statement, specifying a value from 0 through 4,294,967,295 ( $2^{32} - 1$ ).

There are several reasons you might want to prefer one path over another. For example, compared to other paths, one path might be less expensive to use, might have higher bandwidth, or might be more stable.

[Figure 15 on page 80](#) shows a typical network with internal peer sessions and multiple exit points to a neighboring AS.

**Figure 15: Typical Network with IBGP Sessions and Multiple Exit Points**



To reach Device R4, Device R1 can take a path through either Device R2 or Device R3. By default, the local preference is 100 for either route. When the local preferences are equal, Junos OS has rules for breaking the tie and choosing a path. (See [“Understanding BGP](#)

[Path Selection” on page 8.](#)) In this example, the active route is through Device R2 because the router ID of Device R2 is lower than the router ID of Device R3. The following example shows how to override the default behavior with an explicit setting for the local preference. The example configures a local preference of 300 on Device R3, thereby making Device R3 the preferred path to reach Device R4.

### Configuration

- [Configuring Device R1 on page 82](#)
- [Configuring Device R2 on page 84](#)
- [Configuring Device R3 on page 86](#)
- [Configuring Device R4 on page 89](#)

**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 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1

```

**Device R2**

```

set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1

```

**Device R3**

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

**Device R4**

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3
set protocols bgp group external neighbor 24.24.24.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

### *Configuring Device R1*

**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 Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24

[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```
2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
```

```

user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2

```

4. Configure a policy that accepts direct routes.



**NOTE:** Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept

```

5. Configure the router ID and autonomous system (AS) number.

```

[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1

```

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

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 12.12.12.1/24;
    }
  }
}
fe-1/2/1 {
  unit 2 {
    family inet {
      address 13.13.13.1/24;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.1.1/32;
    }
  }
}

```

```
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.1.1;
    export send-direct;
    neighbor 192.168.2.1;
    neighbor 192.168.3.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/1.2;
  }
}

user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;
```

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

### ***Configuring Device R2***

**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 Device R2:

1. Configure the interfaces.  
  
[edit interfaces fe-1/2/0 unit 3]  
user@R2# set family inet address 12.12.12.21/24  
  
[edit interfaces fe-1/2/1 unit 4]  
user@R2# set family inet address 24.24.24.2/24  
  
[edit interfaces lo0 unit 2]  
user@R2# set family inet address 192.168.2.1/32
2. Configure BGP.  
  
[edit protocols bgp group internal]  
user@R2# set type internal

```

user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1

```

```

[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4

```

4. Configure a policy that accepts direct routes.



**NOTE:** Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
user@R2# set then accept

```

5. Configure the router ID and autonomous system (AS) number.

```

[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1

```

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

```

user@R2# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 12.12.12.2/24;
    }
  }
}
fe-1/2/1 {
  unit 4 {
    family inet {
      address 24.24.24.2/24;
    }
  }
}
lo0 {

```

```
unit 2 {
  family inet {
    address 192.168.2.1/32;
  }
}

user@R2# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R2# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.2.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.3.1;
  }
  group external {
    type external;
    export send-direct;
    peer-as 4;
    neighbor 24.24.24.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
    interface fe-1/2/0.3;
    interface fe-1/2/1.4;
  }
}

user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

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

### ***Configuring Device R3***

**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 Device R3:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24
```

```
[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24
```

```
[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1
```

```
[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.



**NOTE:** Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

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

```
user@R3# show interfaces
fe-1/2/0 {
  unit 5 {
```

```
        family inet {
            address 13.13.13.3/24;
        }
    }
}
fe-1/2/1 {
    unit 6 {
        family inet {
            address 34.34.34.3/24;
        }
    }
}
lo0 {
    unit 3 {
        family inet {
            address 192.168.3.1/32;
        }
    }
}

user@R3# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R3# show protocols
bgp {
    group internal {
        type internal;
        local-address 192.168.3.1;
        export send-direct;
        neighbor 192.168.1.1;
        neighbor 192.168.2.1;
    }
    group external {
        type external;
        export send-direct;
        peer-as 4;
        neighbor 34.34.34.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.3 {
            passive;
        }
        interface fe-1/2/0.5;
        interface fe-1/2/1.6;
    }
}

user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;
```

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

### Configuring Device R4

**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 Device R4:

1. Configure the interfaces.  

```
[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
```
2. Configure BGP.  

```
[edit protocols bgp group external]
user@R4# set type external
user@R4# set export send-direct
user@R4# set peer-as 123
user@R4# set neighbor 34.34.34.3
user@R4# set neighbor 24.24.24.2
```
3. Configure a policy that accepts direct routes.



**NOTE:** Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

- ```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```
4. Configure the router ID and autonomous system (AS) number.  

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

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

```
user@R4# show interfaces
fe-1/2/0 {
  unit 7 {
```

```
        family inet {
            address 24.24.24.4/24;
        }
    }
}
fe-1/2/1 {
    unit 8 {
        family inet {
            address 34.34.34.4/24;
        }
    }
}
lo0 {
    unit 4 {
        family inet {
            address 192.168.4.1/32;
        }
    }
}

user@R4# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R4# show protocols
bgp {
    group external {
        type external;
        export send-direct;
        peer-as 123;
        neighbor 34.34.34.3;
        neighbor 24.24.24.2;
    }
}

user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

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

---

## Verification

Confirm that the configuration is working properly.

- [Checking the Active Path From Device R1 to Device R4 on page 90](#)
- [Altering the Local Preference to Change the Path Selection on page 91](#)
- [Rechecking the Active Path From Device R1 to Device R4 on page 91](#)

### ***Checking the Active Path From Device R1 to Device R4***

**Purpose** Verify that the active path from Device R1 to Device R4 goes through Device R2.

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

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 18 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24      [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24      [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24      [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24      [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32     [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32     [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:05:14, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1
                  [BGP/170] 00:05:14, localpref 100, from 192.168.3.1
                  AS path: 4 I
                  > to 13.13.13.3 via fe-1/2/1.2
```

**Meaning** The asterisk (\*) shows that the preferred path is through Device R2. In the default configuration, Device R2 has a lower router ID than Device R3. The router ID is controlling the path selection.

### *Altering the Local Preference to Change the Path Selection*

**Purpose** Change the path so that it goes through Device R3.

**Action** From configuration mode, enter the **set local-preference 300** command.

```
[edit protocols bgp group internal]
user@R3# set local-preference 300
user@R3# commit
```

### *Rechecking the Active Path From Device R1 to Device R4*

**Purpose** Verify that the active path from Device R1 to Device R4 goes through Device R3.

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

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 17 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24      [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24      [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
```

```

                AS path: I
                > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24   [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                AS path: I
                > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24   [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                AS path: I
                > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32  [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                AS path: I
                > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32  [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                AS path: I
                > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32  * [BGP/170] 00:00:21, localpref 300, from 192.168.3.1
                AS path: 4 I
                > to 13.13.13.3 via fe-1/2/1.2
```

**Meaning** The asterisk (\*) shows that the preferred path is through Device R3. In the altered configuration, Device R3 has a higher local preference than Device R2. The local preference is controlling the path selection.

**Related Documentation**

- [Examples: Configuring Internal BGP Peering on page 56](#)
- [BGP Configuration Overview on page 12](#)

---

## Examples: Configuring BGP MED

- [Understanding the MED Attribute on page 92](#)
- [Example: Configuring the MED Attribute Directly on page 95](#)
- [Example: Configuring the MED Using Route Filters on page 107](#)
- [Example: Configuring the MED Using Communities on page 120](#)
- [Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates on page 121](#)

### Understanding the MED Attribute

The BGP multiple exit discriminator (MED, or MULTI\_EXIT\_DISC) is a non-transitive attribute, meaning that it is not propagated throughout the Internet, but only to adjacent autonomous systems (ASs). The MED attribute is optional, meaning that it is not always sent with the BGP updates. The purpose of MED is to influence how other ASs enter your AS to reach a certain prefix.

The MED attribute has a value that is referred to as a *metric*. If all other factors in determining an exit point are equal, the exit point with the lowest metric is preferred.

If a MED is received over an external BGP link, it is propagated over internal links to other BGP-enabled devices within the AS.

BGP update messages include a MED metric if the route was learned from BGP and already had a MED metric associated with it, or if you configure the MED metric in the configuration file.

A MED metric is advertised with a route according to the following general rules:

- A more specific metric overrides a less specific metric. That is, a group-specific metric overrides a global BGP metric, and a peer-specific metric overrides a global BGP or group-specific metric.
- A metric defined with a routing policy overrides a metric defined with the **metric-out** statement.
- If any metric is defined, it overrides a metric received in a route.
- If the received route does not have an associated MED metric, and if you do not explicitly configure a metric value, no metric is advertised. When you do not explicitly configure a metric value, the MED value is equivalent to zero (0) when advertising an active route.

Because the AS path rather than the number of hops between hosts is the primary criterion for BGP route selection, an AS with multiple connections to a peer AS can have multiple equivalent AS paths. When the routing table contains two routes to the same host in a neighboring AS, a MED metric assigned to each route can determine which to include in the forwarding table. The MED metric you assign can force traffic through a particular exit point in an AS.

Figure 16 on page 93 illustrates how MED metrics are used to determine route selection.

Figure 16: Default MED Example

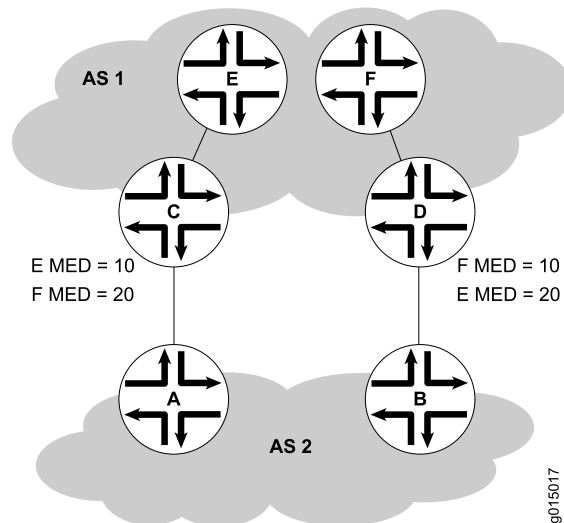


Figure 16 on page 93 shows AS 1 and AS 2 connected by two separate BGP links to Routers C and D. Host E in AS 1 is located nearer to Router C. Host F, also in AS 1, is located nearer to Router D. Because the AS paths are equivalent, two routes exist for each host, one through Router C and one through Router D. To force all traffic destined for Host E through Router C, the network administrator for AS 1 assigns a MED metric for each router to Host E at its exit point. A MED metric of 10 is assigned to the route to Host E through Router C, and a MED metric of 20 is assigned to the route to Host E through Router D. BGP routers in AS 2 select the route with the lower MED metric for the forwarding table.

By default, only the MEDs of routes that have the same peer ASs are compared. However, you can configure the routing table path selection options listed in [Table 3 on page 94](#) to compare MEDs in different ways. The MED options are not mutually exclusive and can be configured in combination or independently. For the MED options to take effect, you must configure them uniformly all through your network. The MED option or options you configure determine the route selected. Thus we recommend that you carefully evaluate your network for preferred routes before configuring the MED options.

**Table 3: MED Options for Routing Table Path Selection**

| Option (Name)                                                                   | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Use                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Always comparing MEDs ( <b>always-compare-med</b> )                             | Ensures that the MEDs for paths from peers in different ASs are always compared in the route selection process.                                                                                                                                                                                                                                                                                                                                                                                                                      | Useful when all enterprises participating in a network agree on a uniform policy for setting MEDs. For example, in a network shared by two ISPs, both must agree that a certain path is the better path to configure the MED values correctly. |
| Adding IGP cost to MED ( <b>med-plus-igp</b> )                                  | <p>Before comparing MED values for path selection, adds to the MED the cost of the IGP route to the BGP next-hop destination.</p> <p>This option replaces the MED value for the router, but does not affect the IGP metric comparison. As a result, when multiple routes have the same value after the MED-plus-IPG comparison, and route selection continues, the IGP route metric is also compared, even though it was added to the MED value and compared earlier in the selection process.</p>                                   | Useful when the downstream AS requires the complete cost of a certain route that is received across multiple ASs.                                                                                                                              |
| Applying Cisco IOS nondeterministic behavior ( <b>cisco-non-deterministic</b> ) | <p>Specifies the nondeterministic behavior of the Cisco IOS software:</p> <ul style="list-style-type: none"> <li>The active path is always first. All nonactive but eligible paths follow the active path and are maintained in the order in which they were received. Ineligible paths remain at the end of the list.</li> <li>When a new path is added to the routing table, path comparisons are made among all routes, including those paths that must never be selected because they lose the MED tie-breaking rule.</li> </ul> | We recommend that you do not configure this option, because the nondeterministic behavior sometimes prevents the system from properly comparing the MEDs between paths.                                                                        |

## Example: Configuring the MED Attribute Directly

This example shows how to configure a multiple exit discriminator (MED) metric to advertise in BGP update messages.

- [Requirements on page 95](#)
- [Overview on page 95](#)
- [Configuration on page 96](#)
- [Verification on page 106](#)

### Requirements

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

### Overview

To directly configure a MED metric to advertise in BGP update messages, include the **metric-out** statement:

**metric-out** (*metric* | *minimum-igp offset* | *igp delay-med-update* | *offset*);

**metric** is the primary metric on all routes sent to peers. It can be a value in the range from 0 through 4,294,967,295 ( $2^{32} - 1$ ).

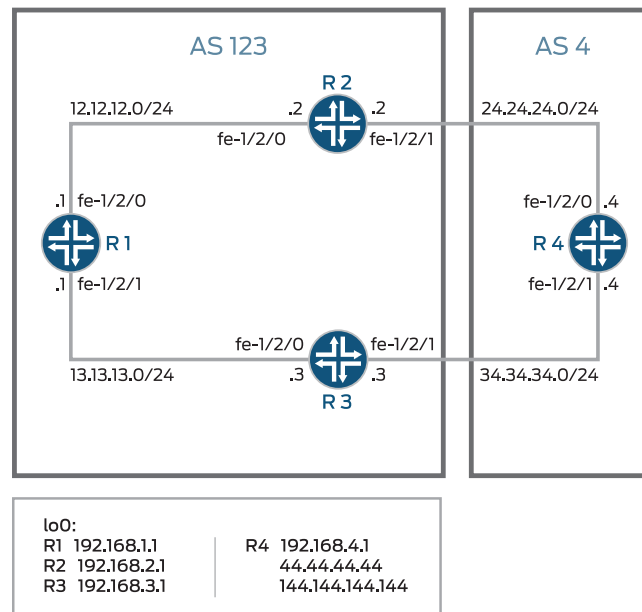
The following optional settings are also supported:

- **minimum-igp**—Sets the metric to the minimum metric value calculated in the interior gateway protocol (IGP) to get to the BGP next hop. If a newly calculated metric is greater than the minimum metric value, the metric value remains unchanged. If a newly calculated metric is lower, the metric value is lowered to that value.
- **igp**—Sets the metric to the most recent metric value calculated in the IGP to get to the BGP next hop.
- **delay-med-update**—Delays sending MED updates when the MED value increases. Include the **delay-med-update** statement when you configure the **igp** statement. The default interval to delay sending updates, unless the MED is lower or another attribute associated with the route has changed is 10 minutes. Include the **med-igp-update-interval minutes** statement at the **[edit routing-options]** hierarchy level to modify the default interval.
- **offset**—Specifies a value for **offset** to increase or decrease the metric that is used from the metric value calculated in the IGP. The metric value is offset by the value specified. The metric calculated in the IGP (by specifying either **igp** or **igp-minimum**) is increased if the **offset** value is positive. The metric calculated in the IGP (by specifying either **igp** or **igp-minimum**) is decreased if the **offset** value is negative.

**offset** can be a value in the range from  $-2^{31}$  through  $2^{31} - 1$ . Note that the adjusted metric can never go below 0 or above  $2^{32} - 1$ .

Figure 17 on page 96 shows a typical network with internal peer sessions and multiple exit points to a neighboring autonomous system (AS).

Figure 17: Typical Network with IBGP Sessions and Multiple Exit Points



Device R4 has multiple loopback interfaces configured to simulate advertised prefixes. The extra loopback interface addresses are 44.44.44.44/32 and 144.144.144.144/32. This example shows how to configure Device R4 to advertise a MED value of 30 to Device R3 and a MED value of 20 to Device R2. This causes all of the devices in AS 123 to prefer the path through Device R2 to reach AS 4.

### Configuration

- [Configuring Device R1 on page 98](#)
- [Configuring Device R2 on page 100](#)
- [Configuring Device R3 on page 102](#)
- [Configuring Device R4 on page 104](#)

#### 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 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
```

```
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1
```

Device R2

```
set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1
```

Device R3

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

Device R4

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set interfaces lo0 unit 4 family inet address 44.44.44.44/32
set interfaces lo0 unit 4 family inet address 144.144.144.144/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3 metric-out 30
set protocols bgp group external neighbor 24.24.24.2 metric-out 20
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
```

```
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

### *Configuring Device R1*

**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 Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24
```

```
[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24
```

```
[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

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

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```

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

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 12.12.12.1/24;
    }
  }
}
fe-1/2/1 {
  unit 2 {
    family inet {
      address 13.13.13.1/24;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.1.1/32;
    }
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.1.1;
    export send-direct;
    neighbor 192.168.2.1;
    neighbor 192.168.3.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/1.2;
  }
}

user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;

```

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

### Configuring Device R2

**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 Device R2:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]  
user@R2# set family inet address 12.12.12.21/24
```

```
[edit interfaces fe-1/2/1 unit 4]  
user@R2# set family inet address 24.24.24.2/24
```

```
[edit interfaces lo0 unit 2]  
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]  
user@R2# set type internal  
user@R2# set local-address 192.168.2.1  
user@R2# set export send-direct  
user@R2# set neighbor 192.168.1.1  
user@R2# set neighbor 192.168.3.1
```

```
[edit protocols bgp group external]  
user@R2# set type external  
user@R2# set export send-direct  
user@R2# set peer-as 4  
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]  
user@R2# set interface lo0.2 passive  
user@R2# set interface fe-1/2/0.3  
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

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

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]  
user@R2# set autonomous-system 123  
user@R2# set router-id 192.168.2.1
```

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

```

user@R2# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 12.12.12.2/24;
    }
  }
}
fe-1/2/1 {
  unit 4 {
    family inet {
      address 24.24.24.2/24;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.2.1/32;
    }
  }
}

user@R2# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R2# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.2.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.3.1;
  }
  group external {
    type external;
    export send-direct;
    peer-as 4;
    neighbor 24.24.24.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
  }
}

```

```
}
interface fe-1/2/0.3;
interface fe-1/2/1.4;
}
}

user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

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

### **Configuring Device R3**

**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 Device R3:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24
```

```
[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24
```

```
[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1
```

```
[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

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

```
user@R3# show interfaces
fe-1/2/0 {
  unit 5 {
    family inet {
      address 13.13.13.3/24;
    }
  }
}
fe-1/2/1 {
  unit 6 {
    family inet {
      address 34.34.34.3/24;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.3.1/32;
    }
  }
}

user@R3# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R3# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.3.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.2.1;
  }
  group external {
```

```

        type external;
        export send-direct;
        peer-as 4;
        neighbor 34.34.34.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.3 {
            passive;
        }
        interface fe-1/2/0.5;
        interface fe-1/2/1.6;
    }
}

user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;

```

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

### Configuring Device R4

**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 Device R4:

1. Configure the interfaces.

```

[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

```

```

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

```

```

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
user@R4# set family inet address 44.44.44.44/32
user@R4# set family inet address 144.144.144.144/32

```

Device R4 has multiple loopback interface addresses to simulate advertised prefixes.

2. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept

```

3. Configure BGP.

```

[edit protocols bgp group external]
user@R4# set type external

```

```

user@R4# set export send-direct
user@R4# set peer-as 123

```

4. Configure a MED value of 30 for neighbor Device R3, and a MED value of 20 for neighbor Device R2.

```

[edit protocols bgp group external]
user@R4# set neighbor 34.34.34.3 metric-out 30
user@R4# set neighbor 24.24.24.2 metric-out 20

```

This configuration causes autonomous system (AS) 123 (of which Device R1, Device R2, and Device R3 are members) to prefer the path through Device R2 to reach AS 4.

5. Configure the router ID and AS number.

```

[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1

```

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

```

user@R4# show interfaces
fe-1/2/0 {
  unit 7 {
    family inet {
      address 24.24.24.4/24;
    }
  }
}
fe-1/2/1 {
  unit 8 {
    family inet {
      address 34.34.34.4/24;
    }
  }
}
lo0 {
  unit 4 {
    family inet {
      address 192.168.4.1/32;
      address 44.44.44.44/32;
      address 144.144.144.144/32;
    }
  }
}

user@R4# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

```

```

user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 34.34.34.3 {
      metric-out 30;
    }
    neighbor 24.24.24.2 {
      metric-out 20;
    }
  }
}

user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;

```

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

## Verification

Confirm that the configuration is working properly.

- [Checking the Active Path From Device R1 to Device R4 on page 106](#)
- [Verifying That Device R4 Is Sending Its Routes Correctly on page 107](#)

### *Checking the Active Path From Device R1 to Device R4*

**Purpose** Verify that the active path goes through Device R2.

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

```

user@R1> show route protocol bgp
inet.0: 13 destinations, 19 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24      [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24      [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24      [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24      [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
44.44.44.44/32     *[BGP/170] 01:41:11, MED 20, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1
144.144.144.144/32 *[BGP/170] 00:08:13, MED 20, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1
192.168.2.1/32     [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                  AS path: I

```

```

192.168.3.1/32      > to 12.12.12.2 via fe-1/2/0.1
                  [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                  AS path: I
192.168.4.1/32      > to 13.13.13.3 via fe-1/2/1.2
                  *[BGP/170] 01:41:11, MED 20, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1

```

**Meaning** The asterisk (\*) shows that the preferred path is through Device R2. The reason for the path selection is listed as MED 20.

### *Verifying That Device R4 Is Sending Its Routes Correctly*

**Purpose** Make sure that Device R4 is sending update messages with a value of 20 to Device R2 and a value of 30 to Device R3.

**Action** From operational mode, enter the **show route advertising-protocol bgp 24.24.24.2** command.

```

user@R4> show route advertising-protocol bgp 24.24.24.2
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref   AS path
* 24.24.24.0/24      Self              20                I
* 34.34.34.0/24      Self              20                I
* 44.44.44.44/32     Self              20                I
* 144.144.144.144/32 Self              20                I
* 192.168.4.1/32     Self              20                I

user@R4> show route advertising-protocol bgp 34.34.34.3
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref   AS path
* 24.24.24.0/24      Self              30                I
* 34.34.34.0/24      Self              30                I
* 44.44.44.44/32     Self              30                I
* 144.144.144.144/32 Self              30                I
* 192.168.4.1/32     Self              30                I

```

**Meaning** The MED column shows that Device R4 is sending the correct MED values to its two external BGP (EBGP) neighbors.

## Example: Configuring the MED Using Route Filters

This example shows how to configure a policy that uses route filters to modify the multiple exit discriminator (MED) metric to advertise in BGP update messages.

- [Requirements on page 107](#)
- [Overview on page 108](#)
- [Configuration on page 108](#)
- [Verification on page 119](#)

### Requirements

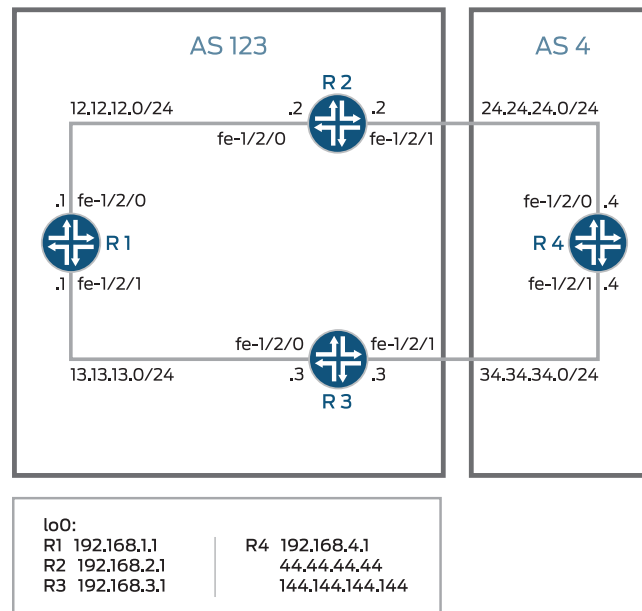
No special configuration beyond device initialization is required before you configure this example.

## Overview

To configure a route-filter policy that modifies the advertised MED metric in BGP update messages, include the **metric** statement in the policy action.

Figure 18 on page 108 shows a typical network with internal peer sessions and multiple exit points to a neighboring autonomous system (AS).

Figure 18: Typical Network with IBGP Sessions and Multiple Exit Points



Device R4 has multiple loopback interfaces configured to simulate advertised prefixes. The extra loopback interface addresses are 44.44.44.44/32 and 144.144.144.144/32. This example shows how to configure Device R4 to advertise a MED value of 30 to Device R3 for all routes except 144.144.144.144. For 144.144.144.144, a MED value of 10 is advertised to Device 3. A MED value of 20 is advertised to Device R2, regardless of the route prefix.

## 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 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1

```

```

set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1

```

Device R2

```

set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1

```

Device R3

```

set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1

```

Device R4

```

set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set interfaces lo0 unit 4 family inet address 44.44.44.44/32
set interfaces lo0 unit 4 family inet address 144.144.144.144/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3 export med-10

```

```
set protocols bgp group external neighbor 34.34.34.3 export med-30
set protocols bgp group external neighbor 24.24.24.2 metric-out 20
set policy-options policy-statement med-10 from route-filter 144.144.144.144/32 exact
set policy-options policy-statement med-10 then metric 10
set policy-options policy-statement med-10 then accept
set policy-options policy-statement med-30 from route-filter 0.0.0.0/0 longer
set policy-options policy-statement med-30 then metric 30
set policy-options policy-statement med-30 then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

### *Configuring Device R1*

**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 Device R1:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24
```

```
[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24
```

```
[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

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

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 12.12.12.1/24;
    }
  }
}
fe-1/2/1 {
  unit 2 {
    family inet {
      address 13.13.13.1/24;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.1.1;
    export send-direct;
    neighbor 192.168.2.1;
    neighbor 192.168.3.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/1.2;
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
```

```
        then accept;
    }
}

user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;
```

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

### **Configuring Device R2**

**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 Device R2:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R2# set family inet address 12.12.12.21/24
```

```
[edit interfaces fe-1/2/1 unit 4]
user@R2# set family inet address 24.24.24.2/24
```

```
[edit interfaces lo0 unit 2]
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R2# set type internal
user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1
```

```
[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
```

```
user@R2# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 12.12.12.2/24;
    }
  }
}
fe-1/2/1 {
  unit 4 {
    family inet {
      address 24.24.24.2/24;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.2.1/32;
    }
  }
}
```

```
user@R2# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.2.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.3.1;
  }
  group external {
    type external;
    export send-direct;
    peer-as 4;
    neighbor 24.24.24.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
  }
}
```

```
    }  
    interface fe-1/2/0.3;  
    interface fe-1/2/1.4;  
  }  
}  
  
user@R2# show policy-options  
policy-statement send-direct {  
  term 1 {  
    from protocol direct;  
    then accept;  
  }  
}  
  
user@R2# show routing-options  
autonomous-system 123;  
router-id 192.168.2.1;
```

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

### ***Configuring Device R3***

**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 Device R3:

1. Configure the device interfaces.  
  
[edit interfaces fe-1/2/0 unit 5]  
user@R3# set family inet address 13.13.13.3/24  
  
[edit interfaces fe-1/2/1 unit 6]  
user@R3# set family inet address 34.34.34.3/24  
  
[edit interfaces lo0 unit 3]  
user@R3# set family inet address 192.168.3.1/32
2. Configure BGP.  
  
[edit protocols bgp group internal]  
user@R3# set type internal  
user@R3# set local-address 192.168.3.1  
user@R3# set export send-direct  
user@R3# set neighbor 192.168.1.1  
user@R3# set neighbor 192.168.2.1  
  
[edit protocols bgp group external]  
user@R3# set type external  
user@R3# set export send-direct  
user@R3# set peer-as 4  
user@R3# set neighbor 34.34.34.4
3. Configure OSPF.  
  
[edit protocols ospf area 0.0.0.0]

```

user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6

```

4. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept

```

5. Configure the router ID and autonomous system (AS) number.

```

[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1

```

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

```

user@R3# show interfaces
fe-1/2/0 {
  unit 5 {
    family inet {
      address 13.13.13.3/24;
    }
  }
}
fe-1/2/1 {
  unit 6 {
    family inet {
      address 34.34.34.3/24;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.3.1/32;
    }
  }
}

```

```

user@R3# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.3.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.2.1;
  }
  group external {

```

```

        type external;
        export send-direct;
        peer-as 4;
        neighbor 34.34.34.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.3 {
            passive;
        }
        interface fe-1/2/0.5;
        interface fe-1/2/1.6;
    }
}

user@R3# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;

```

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

### Configuring Device R4

**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 Device R4:

1. Configure the device interfaces.

```

[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

```

```

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

```

```

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
user@R4# set family inet address 44.44.44.44/32
user@R4# set family inet address 144.144.144.144/32

```

Device R4 has multiple loopback interface addresses to simulate advertised prefixes.

2. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```

3. Configure BGP.

```
[edit protocols bgp group external]
user@R4# set type external
user@R4# set export send-direct
user@R4# set peer-as 123
```

4. Configure the two MED policies.

```
[edit policy-options]
set policy-statement med-10 from route-filter 144.144.144.144/32 exact
set policy-statement med-10 then metric 10
set policy-statement med-10 then accept
```

```
set policy-statement med-30 from route-filter 0.0.0.0/0 longer
set policy-statement med-30 then metric 30
set policy-statement med-30 then accept
```

5. Configure the two EBGP neighbors, applying the two MED policies to Device R3, and a MED value of 20 to Device R2.

```
[edit protocols bgp group external]
user@R4# set neighbor 34.34.34.3 export med-10
user@R4# set neighbor 34.34.34.3 export med-30
user@R4# set neighbor 24.24.24.2 metric-out 20
```

6. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

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

```
user@R4# show interfaces
fe-1/2/0 {
  unit 7 {
    family inet {
      address 24.24.24.4/24;
    }
  }
}
fe-1/2/1 {
  unit 8 {
    family inet {
      address 34.34.34.4/24;
    }
  }
}
lo0 {
```

```
unit 4 {
  family inet {
    address 192.168.4.1/32;
    address 44.44.44.44/32;
    address 144.144.144.144/32;
  }
}

user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 24.24.24.2 {
      metric-out 20;
    }
    neighbor 34.34.34.3 {
      export [ med-10 med-30 ];
    }
  }
}

user@R4# show policy-options
policy-statement med-10 {
  from {
    route-filter 144.144.144.144/32 exact;
  }
  then {
    metric 10;
    accept;
  }
}
policy-statement med-30 {
  from {
    route-filter 0.0.0.0/0 longer;
  }
  then {
    metric 30;
    accept;
  }
}
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

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

## Verification

Confirm that the configuration is working properly.

- [Checking the Active Path from Device R1 to Device R4 on page 119](#)
- [Verifying That Device R4 Is Sending Its Routes Correctly on page 119](#)

### *Checking the Active Path from Device R1 to Device R4*

**Purpose** Verify that the active path goes through Device R2.

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

```
user@R1> show route protocol bgp
inet.0: 13 destinations, 19 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24      [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24      [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24      [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24      [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
44.44.44.44/32     *[BGP/170] 00:06:03, MED 20, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1
144.144.144.144/32 *[BGP/170] 00:06:03, MED 10, localpref 100, from 192.168.3.1
                  AS path: 4 I
                  > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32     [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                  AS path: I
                  > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32     [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                  AS path: I
                  > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:06:03, MED 20, localpref 100, from 192.168.2.1
                  AS path: 4 I
                  > to 12.12.12.2 via fe-1/2/0.1
```

**Meaning** The output shows that the preferred path to the routes advertised by Device R4 is through Device R2 for all routes except 144.144.144.144/32. For 144.144.144.144/32, the preferred path is through Device R3.

### *Verifying That Device R4 Is Sending Its Routes Correctly*

**Purpose** Make sure that Device R4 is sending update messages with a value of 20 to Device R2 and a value of 30 to Device R3.

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

```
user@R4> show route advertising-protocol bgp 24.24.24.2
```

```
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref   AS path
* 24.24.24.0/24         Self                20              I
* 34.34.34.0/24         Self                20              I
* 44.44.44.44/32        Self                20              I
* 144.144.144.144/32    Self                20              I
* 192.168.4.1/32        Self                20              I
```

```
user@R4> show route advertising-protocol bgp 34.34.34.3
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref   AS path
* 24.24.24.0/24         Self                30              I
* 34.34.34.0/24         Self                30              I
* 44.44.44.44/32        Self                30              I
* 144.144.144.144/32    Self                10              I
* 192.168.4.1/32        Self                30              I
```

**Meaning** The MED column shows that Device R4 is sending the correct MED values to its two EBGp neighbors.

### Example: Configuring the MED Using Communities

Set the multiple exit discriminator (MED) metric to 20 for all routes from a particular community.

```
[edit]
routing-options {
  router-id 10.0.0.1;
  autonomous-system 23;
}
policy-options {
  policy-statement from-otago {
    from community otago;
    then metric 20;
  }
  community otago members [56:2379 23:46944];
}
protocols {
  bgp {
    import from-otago;
    group 23 {
      type external;
      peer-as 56;
      neighbor 192.168.0.1 {
        traceoptions {
          file bgp-log-peer;
          flag packets;
        }
        log-updown;
      }
    }
  }
}
```

## Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates

This example shows how to associate the multiple exit discriminator (MED) path attribute with the interior gateway protocol (IGP) metric, and configure a timer to delay update of the MED attribute.

- [Requirements on page 121](#)
- [Overview on page 121](#)
- [Configuration on page 123](#)
- [Verification on page 129](#)

### Requirements

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

### Overview

BGP can be configured to advertise the MED attribute for a route based on the IGP distance of its internal BGP (IBGP) route next-hop. The IGP metric enables internal routing to follow the shortest path according to the administrative setup. In some deployments, it might be ideal to communicate IGP shortest-path knowledge to external BGP (EBGP) peers in a neighboring autonomous system (AS). This allows those EBGP peers to forward traffic into your AS using the shortest paths possible.

Routes learned from an EBGP peer usually have a next hop on a directly connected interface, and thus the IGP value is equal to zero. Zero is the value advertised. The IGP metric is a nonzero value when a BGP peer sends third-party next hops that require the local system to perform next-hop resolution—IBGP configurations, configurations within confederation peers, or EBGP configurations that include the **multihop** command. In these scenarios, it might make sense to associate the MED value with the IGP metric by including the **metric-out minimum-igp** or **metric-out igp** option.

The drawback of associating the MED with the IGP metric is the risk of excessive route advertisements when there are IGP instabilities in the network. Configuring a delay for the MED update provides a mechanism to reduce route advertisements in such scenarios. The delay works by slowing down MED updates when the IGP metric for the next hop changes. The approach uses a timer to periodically advertise MED updates. When the timer expires, the MED attribute for routes with **metric-out igp delay-updates** configured is updated to the current IGP metric of the next hop. The BGP-enabled device sends out advertisements for routes for which the MED attribute has changed.

The **delay-updates** option identifies the BGP groups (or peers) for which the MED updates must be suppressed. The time for advertising MED updates is set to 10 minutes by default. You can increase the interval up to 600 minutes by including the **med-igp-update-interval** statement in the **routing-options** configuration.



NOTE: If you have nonstop active routing (NSR) enabled and a switchover occurs, the delayed MED updates might be advertised as soon as the switchover occurs.

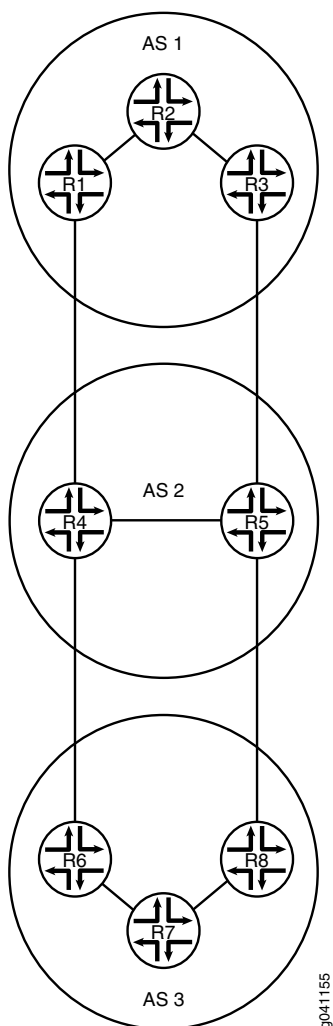
When you configure the **metric-out igp** option, the IGP metric directly tracks the IGP cost to the IBGP peer. When the IGP cost goes down, so does the advertised MED value. Conversely, when the IGP cost goes up, the MED value goes up as well.

When you configure the **metric-out minimum-igp** option, the advertised MED value changes only when the IGP cost to the IBGP peer goes down. An increase in the IGP cost does not affect the MED value. The router monitors and remembers the lowest IGP cost until the routing process (rpd) is restarted. The BGP peer sends an update only if the MED is lower than the previously advertised value or another attribute associated with the route has changed, or if the BGP peer is responding to a refresh route request.

This example uses the **metric** statement in the OSPF configuration to demonstrate that when the IGP metric changes, the MED also changes after the configured delay interval. The OSPF metric can range from 1 through 65,535.

[Figure 19 on page 123](#) shows the sample topology.

Figure 19: Topology for Delaying the MED Update



In this example, the MED value advertised by Device R1 is associated with the IGP running in AS 1. The MED value advertised by Device R1 impacts the decisions of the neighboring AS (AS 2) when AS 2 is forwarding traffic into AS 1.

### Configuration

- [Configuring Device R1 on page 127](#)

#### 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 2 description R1->R2
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.1/30
set interfaces fe-1/2/1 unit 7 description R1->R4
set interfaces fe-1/2/1 unit 7 family inet address 172.16.0.1/30
set interfaces lo0 unit 1 family inet address 192.168.0.1/32
  
```

```
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.2
set protocols bgp group internal neighbor 192.168.0.3
set protocols bgp group external type external
set protocols bgp group external metric-out igp delay-med-update
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.2
set protocols ospf area 0.0.0.0 interface fe-1/2/0.2 metric 600
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options med-igp-update-interval 12
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 1
```

Device R2

```
set interfaces fe-1/2/0 unit 1 description R2->R1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 4 description R2->R3
set interfaces fe-1/2/1 unit 4 family inet address 10.0.2.2/30
set interfaces lo0 unit 2 family inet address 192.168.0.2/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.2
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.1
set protocols bgp group internal neighbor 192.168.0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 1
```

Device R3

```
set interfaces fe-1/2/0 unit 3 description R3->R2
set interfaces fe-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces fe-1/2/1 unit 5 description R3->R5
set interfaces fe-1/2/1 unit 5 family inet address 172.16.0.5/30
set interfaces lo0 unit 3 family inet address 192.168.0.3/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.3
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.1
set protocols bgp group internal neighbor 192.168.0.2
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.6
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.3
```

```
set routing-options autonomous-system 1
```

**Device R4**

```

set interfaces fe-1/2/0 unit 8 description R4->R1
set interfaces fe-1/2/0 unit 8 family inet address 172.16.0.2/30
set interfaces fe-1/2/1 unit 9 description R4->R5
set interfaces fe-1/2/1 unit 9 family inet address 10.0.4.1/30
set interfaces fe-1/2/2 unit 13 description R4->R6
set interfaces fe-1/2/2 unit 13 family inet address 172.16.0.9/30
set interfaces lo0 unit 4 family inet address 192.168.0.4/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.4
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.5
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external neighbor 172.16.0.10 peer-as 3
set protocols bgp group external neighbor 172.16.0.1 peer-as 1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.9
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.4
set routing-options autonomous-system 2

```

**Device R5**

```

set interfaces fe-1/2/0 unit 6 description R5->R3
set interfaces fe-1/2/0 unit 6 family inet address 172.16.0.6/30
set interfaces fe-1/2/1 unit 10 description R5->R4
set interfaces fe-1/2/1 unit 10 family inet address 10.0.4.2/30
set interfaces fe-1/2/2 unit 11 description R5->R8
set interfaces fe-1/2/2 unit 11 family inet address 172.16.0.13/30
set interfaces lo0 unit 5 family inet address 192.168.0.5/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.5
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.4
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external neighbor 172.16.0.5 peer-as 1
set protocols bgp group external neighbor 172.16.0.14 peer-as 3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.10
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.5
set routing-options autonomous-system 2

```

**Device R6**

```

set interfaces fe-1/2/0 unit 14 description R6->R4
set interfaces fe-1/2/0 unit 14 family inet address 172.16.0.10/30
set interfaces fe-1/2/1 unit 15 description R6->R7
set interfaces fe-1/2/1 unit 15 family inet address 10.0.6.1/30
set interfaces lo0 unit 6 family inet address 192.168.0.6/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.6
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.7

```

```
set protocols bgp group internal neighbor 192.168.0.8
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.9 peer-as 2
set protocols ospf area 0.0.0.0 interface fe-1/2/1.15
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.6
set routing-options autonomous-system 3
```

Device R7

```
set interfaces fe-1/2/0 unit 16 description R7->R6
set interfaces fe-1/2/0 unit 16 family inet address 10.0.6.2/30
set interfaces fe-1/2/1 unit 17 description R7->R8
set interfaces fe-1/2/1 unit 17 family inet address 10.0.7.2/30
set interfaces lo0 unit 7 family inet address 192.168.0.7/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.7
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.6
set protocols bgp group internal neighbor 192.168.0.8
set protocols ospf area 0.0.0.0 interface fe-1/2/0.16
set protocols ospf area 0.0.0.0 interface fe-1/2/1.17
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.7
set routing-options autonomous-system 3
```

Device R8

```
set interfaces fe-1/2/0 unit 12 description R8->R5
set interfaces fe-1/2/0 unit 12 family inet address 172.16.0.14/30
set interfaces fe-1/2/1 unit 18 description R8->R7
set interfaces fe-1/2/1 unit 18 family inet address 10.0.7.1/30
set interfaces lo0 unit 8 family inet address 192.168.0.8/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.8
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.6
set protocols bgp group internal neighbor 192.168.0.7
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.13 peer-as 2
set protocols ospf area 0.0.0.0 interface fe-1/2/1.18
set protocols ospf area 0.0.0.0 interface lo0.8 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.8
set routing-options autonomous-system 3
```

**Configuring Device R1**

**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 Device R1:

1. Configure the interfaces.

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

```
[edit interfaces fe-1/2/1 unit 7]
user@R1# set description R1->R4
user@R1# set family inet address 172.16.0.1/30
```

```
[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.0.1/32
```

2. Configure IBGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.0.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.0.2
user@R1# set neighbor 192.168.0.3
```

3. Configure EBGP.

```
[edit protocols bgp group external]
user@R1# set type external
user@R1# set export send-direct
user@R1# set peer-as 2
user@R1# set neighbor 172.16.0.2
```

4. Associate the MED value with the IGP metric.

```
[edit protocols bgp group external]
user@R1# set metric-out igp delay-med-update
```

The default for the MED update is 10 minutes when you include the **delay-med-update** option. When you exclude the **delay-med-update** option, the MED update occurs immediately after the IGP metric changes.

5. (Optional) Configure the update interval for the MED update.

```
[edit routing-options]
user@R1# set med-igp-update-interval 12
```

You can configure the interval from 10 minutes through 600 minutes.

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/0.2 metric 600
user@R1# set interface lo0.1 passive
```

The **metric** statement is used here to demonstrate what happens when the IGP metric changes.

7. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

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

8. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set router-id 192.168.0.1
user@R1# set autonomous-system 1
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 2 {
    description R1->R2;
    family inet {
      address 10.0.0.1/30;
    }
  }
}
fe-1/2/1 {
  unit 7 {
    description R1->R4;
    family inet {
      address 172.16.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show protocols
```

```

bgp {
  group internal {
    type internal;
    local-address 192.168.0.1;
    export send-direct;
    neighbor 192.168.0.2;
    neighbor 192.168.0.3;
  }
  group external {
    type external;
    metric-out igp delay-med-update;
    export send-direct;
    peer-as 2;
    neighbor 172.16.0.2;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.2 {
      metric 600;
    }
    interface lo0.1 {
      passive;
    }
  }
}

```

```

user@R1# show routing-options
med-igp-update-interval 12;
router-id 192.168.0.1;
autonomous-system 1;

```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration steps on the other devices in the topology, as needed for your network.

## Verification

Confirm that the configuration is working properly.

- [Checking the BGP Advertisements on page 129](#)
- [Verifying That the MED Value Changes When the OSPF Metric Changes on page 130](#)
- [Testing the minimum-igp Setting on page 130](#)

### Checking the BGP Advertisements

**Purpose** Verify that Device R1 is advertising to Device R4 a BGP MED value that reflects the IGP metric.

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

```

user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                Nexthop        MED      Lc1pref  AS path
* 10.0.0.0/30           Self           0         I         I
* 172.16.0.0/30         Self           0         I         I

```

|                  |      |     |   |
|------------------|------|-----|---|
| * 172.16.0.4/30  | Self | 601 | I |
| * 192.168.0.1/32 | Self | 0   | I |

**Meaning** The 601 value in the MED column shows that the MED value has been updated to reflect the configured OSPF metric.

#### *Verifying That the MED Value Changes When the OSPF Metric Changes*

**Purpose** Make sure that when you raise the OSPF metric to 700, the MED value is updated to reflect this change.

**Action** From configuration mode, enter the **set protocols ospf area 0 interface fe-1/2/0.2 metric 700** command.

```
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 700
user@R1# commit
```

After waiting 12 minutes (the configured delay period), enter the **show route advertising-protocol bgp** command from operational mode.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix            Nexthop      MED      Lclpref  AS path
* 10.0.0.0/30       Self         0         I
* 172.16.0.0/30     Self         0         I
* 172.16.0.4/30     Self         701        I
* 192.168.0.1/32    Self         0         I
```

**Meaning** The 701 value in the MED column shows that the MED value has been updated to reflect the configured OSPF metric.

#### *Testing the minimum-igp Setting*

**Purpose** Change the configuration to use the **minimum-igp** statement instead of the **igp** statement. When you increase the OSPF metric, the MED value remains unchanged, but when you decrease the OSPF metric, the MED value reflects the new OSPF metric.

**Action** From configuration mode, delete the **igp** statement, add the **minimum-igp** statement, and increase the OSPF metric.

```
user@R1# delete protocols bgp group external metric-out igp
user@R1# set protocols bgp group external metric-out minimum-igp
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 800
user@R1# commit
```

From operational mode, enter the **show route advertising-protocol bgp** command to make sure that the MED value does not change.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix            Nexthop      MED      Lclpref  AS path
* 10.0.0.0/30       Self         0         I
* 172.16.0.0/30     Self         0         I
* 172.16.0.4/30     Self         701        I
* 192.168.0.1/32    Self         0         I
```

From configuration mode, decrease the OSPF metric.

```
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 20
user@R1# commit
```

From operational mode, enter the **show route advertising-protocol bgp** command to make sure that the MED value does change.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref    AS path
* 10.0.0.0/30           Self             0         I          I
* 172.16.0.0/30         Self             0         I          I
* 172.16.0.4/30        Self             21        I          I
* 192.168.0.1/32       Self             0         I          I
```

**Meaning** When the **minimum-igp** statement is configured, the MED value changes only when a shorter path is available.

**Related Documentation**

- [Examples: Configuring External BGP Peering on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Examples: Configuring BGP Local AS

- [Understanding the BGP Local AS Attribute on page 131](#)
- [Example: Configuring a Local AS for EBGp Sessions on page 136](#)
- [Example: Configuring a Private Local AS for EBGp Sessions on page 146](#)

### Understanding the BGP Local AS Attribute

When an Internet service provider (ISP) acquires a network that belongs to a different autonomous system (AS), there is no seamless method for moving the BGP peers of the acquired network to the AS of the acquiring ISP. The process of configuring the BGP peers with the new AS number can be time-consuming and cumbersome. Sometimes customers do not want to or are not immediately able to modify their peer arrangements or configuration. During this kind of transition period, it can be useful to configure BGP-enabled devices in the new AS to use the former AS number in BGP updates. This former AS number is called a *local AS*.

Using a local AS number permits the routing devices in an acquired network to appear to belong to the former AS.

For example, ISP A, with an AS of 200, acquires ISP B, with an AS of 250. ISP B has a customer, ISP C, that does not want to change its configuration. After ISP B becomes part of ISP A, a local AS number of 250 is configured for use in EBGp peer sessions with ISP C. Consequently, the local AS number of 250 is either prepended before or used instead of the global AS number of 200 in the AS path used to export routes to direct external peers in ISP C.

If the route is received from an internal BGP (IBGP) peer, the AS path includes the local AS number prepended before the global AS number.

The local AS number is used instead of the global AS number if the route is an external route, such as a static route or an interior gateway protocol (IGP) route that is imported into BGP. If the route is external and you want the global AS number to be included in the AS path, you can apply a routing policy that uses **as-path-expand** or **as-path-prepend**. Use the **as-path-expand** policy action to place the global AS number behind the local AS number. Use the **as-path-prepend** policy action to place the global AS number in front of the local AS number.

For example:

```

user@R2# show policy-options
policy-statement prepend-global {
  term 1 {
    from protocol static;
    then {
      as-path-prepend 200; # or use as-path-expand
    }
  }
}

user@R2# show protocols bgp
group ext {
  export prepend-global;
  type external;
  local-as 250;
  neighbor 10.0.0.1 {
    peer-as 100;
  }
  neighbor 10.1.0.2 {
    peer-as 300;
  }
}

user@R2# show routing-options
static {
  route 1.1.1.1/32 next-hop 10.0.0.1;
}
autonomous-system 200;

user@R3# run show route 1.1.1.1 protocol bgp
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32          *[BGP/170] 00:05:11, localpref 100
                   AS path: 200 250 I, validation-state: unverified
                   > to 10.1.0.1 via 1t-1/2/0.4

```

In a Layer 3 VPN scenario, in which a provider edge (PE) device uses external BGP (EBGP) to peer with a customer edge (CE) device, the **local-as** statement behaves differently than in the non-VPN scenario. In the VPN scenario, the global AS number defined in the master instance is prepended to the AS path by default. To override this behavior, you can configure the **no-prepend-global-as** in the routing-instance BGP configuration on the PE device, as shown here:

```

user@R2# show routing-instances

```

```
red {
  instance-type vrf;
  interface fe-1/2/0.2;
  route-distinguisher 2:1;
  vrf-target target:2:1;
  protocols {
    bgp {
      group toR1 {
        type external;
        peer-as 1;
        local-as 200 no-prepend-global-as;
        neighbor 10.1.1.1;
      }
    }
  }
}
```

The Junos operating system (Junos OS) implementation of the local AS attribute supports the following options:

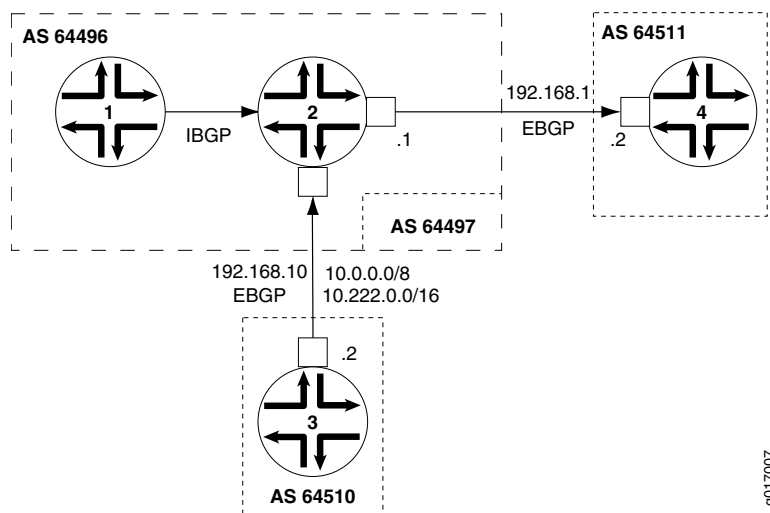
- **Local AS with private option**—When you use the **private** option, the local AS is used during the establishment of the BGP session with an EBGP neighbor but is hidden in the AS path sent to other EBGP peers. Only the global AS is included in the AS path sent to external peers.

The **private** option is useful for establishing local peering with routing devices that remain configured with their former AS or with a specific customer that has not yet modified its peer arrangements. The local AS is used to establish the BGP session with the EBGP neighbor but is hidden in the AS path sent to external peers in another AS.

Include the **private** option so that the local AS is not prepended before the global AS in the AS path sent to external peers. When you specify the **private** option, the local AS is prepended only in the AS path sent to the EBGP neighbor.

For example, in [Figure 20 on page 134](#), Router 1 and Router 2 are in AS 64496, Router 4 is in AS 64511, and Router 3 is in AS 64510. Router 2 formerly belonged to AS 64497, which has merged with another network and now belongs to AS 64496. Because Router 3 still peers with Router 2 using its former AS (64497), Router 2 needs to be configured with a local AS of 64497 in order to maintain peering with Router 3. Configuring a local AS of 64497 permits Router 2 to add AS 64497 when advertising routes to Router 3. Router 3 sees an AS path of 64497 64496 for the prefix 10/8.

**Figure 20: Local AS Configuration**



To prevent Router 2 from adding the local AS number in its announcements to other peers, use the **local-as 64497 private** statement. This statement configures Router 2 to not include local AS 64497 when announcing routes to Router 1 and to Router 4. In this case, Router 4 sees an AS path of 64496 64510 for the prefix 10.222/16.

- **Local AS with alias option**—In Junos OS Release 9.5 and later, you can configure a local AS as an alias. During the establishment of the BGP open session, the AS used in the open message alternates between the local AS and the global AS. If the local AS is used to connect with the EBGP neighbor, then only the local AS is prepended to the AS path when the BGP peer session is established. If the global AS is used to connect with the EBGP neighbor, then only the global AS is prepended to the AS path when the BGP peer session is established. The use of the **alias** option also means that

the local AS is not prepended to the AS path for any routes learned from that EBGp neighbor. Therefore, the local AS remains hidden from other external peers.

Configuring a local AS with the **alias** option is especially useful when you are migrating the routing devices in an acquired network to the new AS. During the migration process, some routing devices might be configured with the new AS while others remain configured with the former AS. For example, it is good practice to start by first migrating to the new AS any routing devices that function as route reflectors. However, as you migrate the route reflector clients incrementally, each route reflector has to peer with routing devices configured with the former AS, as well as peer with routing devices configured with the new AS. To establish local peer sessions, it can be useful for the BGP peers in the network to use both the local AS and the global AS. At the same time, you want to hide this local AS from external peers and use only the global AS in the AS path when exporting routes to another AS. In this kind of situation, configure the **alias** option.

Include the **alias** option to configure the local AS as an alias to the global AS configured at the **[edit routing-options]** hierarchy level. When you configure a local AS as an alias, during the establishment of the BGP open session, the AS used in the open message alternates between the local AS and the global AS. The local AS is prepended to the AS path only when the peer session with an EBGp neighbor is established using that local AS. The local AS is hidden in the AS path sent to any other external peers. Only the global AS is prepended to the AS path when the BGP session is established using the global AS.



**NOTE:** The **private** and **alias** options are mutually exclusive. You cannot configure both options with the same **local-as** statement.

- **Local AS with option not to prepend the global AS**—In Junos OS Release 9.6 and later, you can configure a local AS with the option not to prepend the global AS. Only the local AS is included in the AS path sent to external peers.

Use the **no-prepend-global-as** option when you want to strip the global AS number from outbound BGP updates in a virtual private network (VPN) scenario. This option is useful in a VPN scenario in which you want to hide the global AS from the VPN.

Include the **no-prepend-global-as** option to have the global AS configured at the **[edit routing-options]** hierarchy level removed from the AS path sent to external peers. When you use this option, only the local AS is included in the AS path for the routes sent to a customer edge (CE) device.

- **Number of loops option**—The local AS feature also supports specifying the number of times that detection of the AS number in the AS\_PATH attribute causes the route to be discarded or hidden. For example, if you configure **loops 1**, the route is hidden if the AS number is detected in the path one or more times. This is the default behavior. If you configure **loops 2**, the route is hidden if the AS number is detected in the path two or more times.

For the **loops number** statement, you can configure 1 through 10.



**NOTE:** If you configure the local AS values for any BGP group, the detection of routing loops is performed using both the AS and the local AS values for all BGP groups.

If the local AS for the EBGP or IBGP peer is the same as the current AS, do not use the `local-as` statement to specify the local AS number.

When you configure the local AS within a VRF, this impacts the AS path loop-detection mechanism. All of the `local-as` statements configured on the device are part of a single AS domain. The AS path loop-detection mechanism is based on looking for a matching AS present in the domain.

---

## Example: Configuring a Local AS for EBGP Sessions

This example shows how to configure a local autonomous system (AS) for a BGP peer so that both the global AS and the local AS are used in BGP inbound and outbound updates.

- [Requirements on page 136](#)
- [Overview on page 136](#)
- [Configuration on page 137](#)
- [Verification on page 143](#)

### Requirements

---

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

### Overview

---

Use the **local-as** statement when ISPs merge and want to preserve a customer's configuration, particularly the AS with which the customer is configured to establish a peer relationship. The **local-as** statement simulates the AS number already in place in customer routers, even if the ISP's router has moved to a different AS.

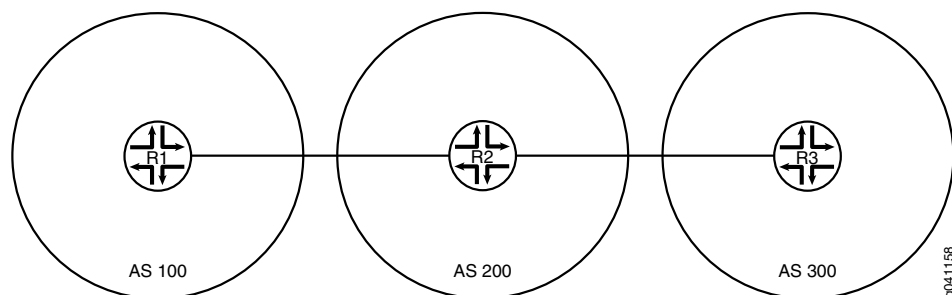
This example shows how to use the **local-as** statement to configure a local AS. The **local-as** statement is supported for BGP at the global, group, and neighbor hierarchy levels.

When you configure the **local-as** statement, you must specify an AS number. You can specify a number from 1 through 4,294,967,295 in plain-number format. In Junos OS Release 9.1 and later, the range for AS numbers is extended to provide BGP support for 4-byte AS numbers as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*. In Junos OS Release 9.3 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>.<16-bit low-order value in decimal>*. For example, the 4-byte AS number of 65,546 in plain-number format is represented as 1.10 in the AS-dot notation format. You can specify a value from 0.0 through 65535.65535 in AS-dot notation format. Junos

OS continues to support 2-byte AS numbers. The 2-byte AS number range is 1 through 65,535 (this is a subset of the 4-byte range).

Figure 21 on page 137 shows the sample topology.

Figure 21: Topology for Configuring the Local AS



In this example, Device R2 formerly belonged to AS 250 and now is in AS 200. Device R1 and Device R3 are configured to peer with AS 250 instead of with the new AS number (AS 200). Device R2 has the new AS number configured with the **autonomous-system 200** statement. To enable the peering sessions to work, the **local-as 250** statement is added in the BGP configuration. Because **local-as 250** is configured, Device R2 includes both the global AS (200) and the local AS (250) in its BGP inbound and outbound updates.

### Configuration

- [Configuring Device R1 on page 138](#)
- [Configuring Device R2 on page 140](#)
- [Configuring Device R3 on page 142](#)

#### 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 192.168.0.1/32
           set protocols bgp group ext type external
           set protocols bgp group ext export send-direct
           set protocols bgp group ext export send-static
           set protocols bgp group ext peer-as 250
           set protocols bgp group ext neighbor 10.0.0.2
           set policy-options policy-statement send-direct term 1 from protocol direct
           set policy-options policy-statement send-direct term 1 then accept
           set policy-options policy-statement send-static term 1 from protocol static
           set policy-options policy-statement send-static term 1 then accept
           set routing-options static route 10.1.0.0/30 next-hop 10.0.0.2
           set routing-options autonomous-system 100

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 3 family inet address 10.1.0.1/30
           set interfaces lo0 unit 2 family inet address 192.168.0.2/32
```

```
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext local-as 250
set protocols bgp group ext neighbor 10.0.0.1 peer-as 100
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options autonomous-system 200
```

**Device R3**

```
set interfaces fe-1/2/0 unit 4 family inet address 10.1.0.2/30
set interfaces lo0 unit 3 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 250
set protocols bgp group ext neighbor 10.1.0.1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.0.0.0/30 next-hop 10.1.0.1
set routing-options autonomous-system 300
```

### *Configuring Device R1*

**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 Device R1:

1. Configure the 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 192.168.0.1/32
2. Configure external BGP (EBGP).  
  
[edit protocols bgp group ext]  
user@R1# set type external  
user@R1# set export send-direct  
user@R1# set export send-static  
user@R1# set peer-as 250  
user@R1# set neighbor 10.0.0.2
3. Configure the routing policy.  
  
[edit policy-options]  
user@R1# set policy-statement send-direct term 1 from protocol direct  
user@R1# set policy-statement send-direct term 1 then accept  
user@R1# set policy-statement send-static term 1 from protocol static  
user@R1# set policy-statement send-static term 1 then accept

4. Configure a static route to the remote network between Device R2 and Device R3.

```
[edit routing-options]
user@R1# set static route 10.1.0.0/30 next-hop 10.0.0.2
```

5. Configure the global AS number.

```
[edit routing-options]
user@R1# set autonomous-system 100
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}

user@R1# show protocols
bgp {
  group ext {
    type external;
    export [ send-direct send-static ];
    peer-as 250;
    neighbor 10.0.0.2;
  }
}

user@R1# show routing-options
static {
```

```
    route 10.1.0.0/30 next-hop 10.0.0.2;
  }
  autonomous-system 100;
```

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

### **Configuring Device R2**

**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 Device R2:

1. Configure the 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 3 family inet address 10.1.0.1/30

user@R2# set lo0 unit 2 family inet address 192.168.0.2/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set export send-direct
user@R2# set export send-static
user@R2# set neighbor 10.0.0.1 peer-as 100
user@R2# set neighbor 10.1.0.2 peer-as 300
```

3. Configure the local autonomous system (AS) number.

```
[edit protocols bgp group ext]
user@R2# set local-as 250
```

4. Configure the global AS number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

5. Configure the routing policy.

```
[edit policy-options]
user@R2# set policy-statement send-direct term 1 from protocol direct
user@R2# set policy-statement send-direct term 1 then accept
user@R2# set policy-statement send-static term 1 from protocol static
user@R2# set policy-statement send-static term 1 then accept
```

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

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

```

        family inet {
            address 10.0.0.2/30;
        }
    }
}
fe-1/2/1 {
    unit 3 {
        family inet {
            address 10.1.0.1/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.0.2/32;
        }
    }
}

user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}

user@R2# show protocols
bgp {
    group ext {
        type external;
        export [ send-direct send-static ];
        local-as 250;
        neighbor 10.0.0.1 {
            peer-as 100;
        }
        neighbor 10.1.0.2 {
            peer-as 300;
        }
    }
}

user@R2# show routing-options
autonomous-system 200;

```

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

**Configuring Device R3**

**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 Device R3:

1. Configure the interfaces.  

```
[edit interfaces]
user@R3# set fe-1/2/0 unit 4 family inet address 10.1.0.2/30

user@R3# set lo0 unit 3 family inet address 192.168.0.3/32
```
2. Configure EBGP.  

```
[edit protocols bgp group ext]
user@R3# set type external
user@R3# set export send-direct
user@R3# set export send-static
user@R3# set peer-as 250
user@R3# set neighbor 10.1.0.1
```
3. Configure the global autonomous system (AS) number.  

```
[edit routing-options]
user@R3# set autonomous-system 300
```
4. Configure a static route to the remote network between Device R1 and Device R2.  

```
[edit routing-options]
user@R3# set static route 10.0.0.0/30 next-hop 10.1.0.1
```
5. Configure the routing policy.  

```
[edit policy-options]
user@R3# set policy-statement send-direct term 1 from protocol direct
user@R3# set policy-statement send-direct term 1 then accept
user@R3# set policy-statement send-static term 1 from protocol static
user@R3# set policy-statement send-static term 1 then accept
```

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

```
user@R3# show interfaces
fe-1/2/0 {
  unit 4 {
    family inet {
      address 10.1.0.2/30;
    }
  }
}
lo0 {
  unit 3 {
```

```

        family inet {
            address 192.168.0.3/32;
        }
    }
}

user@R3# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}

user@R3# show protocols
bgp {
    group ext {
        type external;
        export [ send-direct send-static ];
        peer-as 250;
        neighbor 10.1.0.1;
    }
}

user@R3# show routing-options
static {
    route 10.0.0.0/30 next-hop 10.1.0.1;
}
autonomous-system 300;

```

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

### Verification

Confirm that the configuration is working properly.

- [Checking the Local and Global AS Settings on page 143](#)
- [Checking the BGP Peering Sessions on page 145](#)
- [Verifying the BGP AS Paths on page 145](#)

#### *Checking the Local and Global AS Settings*

**Purpose** Make sure that Device R2 has the local and global AS settings configured.

**Action** From operational mode, enter the **show bgp neighbors** command.

```

user@R2> show bgp neighbors
Peer: 10.0.0.1+179 AS 100      Local: 10.0.0.2+61036 AS 250
Type: External  State: Established  Flags: <Sync>
Last State: OpenConfirm  Last Event: RecvKeepAlive
Last Error: None

```

```

Export: [ send-direct send-static ]
Options: <Preference PeerAS LocalAS Refresh>
Holdtime: 90 Preference: 170 Local AS: 250 Local System AS: 200
Number of flaps: 0
Peer ID: 192.168.0.1      Local ID: 192.168.0.2      Active Holdtime: 90
Keepalive Interval: 30      Peer index: 0
BFD: disabled, down
Local Interface: fe-1/2/0.2
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Stale routes from peer are kept for: 300
Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 100)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        3
  Accepted prefixes:        2
  Suppressed due to damping: 0
  Advertised prefixes:      4
Last traffic (seconds): Received 6    Sent 14    Checked 47
Input messages: Total 258    Updates 3    Refreshes 0    Octets 4969
Output messages: Total 258    Updates 2    Refreshes 0    Octets 5037
Output Queue[0]: 0

Peer: 10.1.0.2+179 AS 300      Local: 10.1.0.1+52296 AS 250
Type: External    State: Established    Flags: <Sync>
Last State: OpenConfirm    Last Event: RecvKeepAlive
Last Error: None
Export: [ send-direct send-static ]
Options: <Preference PeerAS LocalAS Refresh>
Holdtime: 90 Preference: 170 Local AS: 250 Local System AS: 200
Number of flaps: 0
Peer ID: 192.168.0.3      Local ID: 192.168.0.2      Active Holdtime: 90
Keepalive Interval: 30      Peer index: 1
BFD: disabled, down
Local Interface: fe-1/2/1.3
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Stale routes from peer are kept for: 300
Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 300)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        3
  Accepted prefixes:        2

```

```

    Suppressed due to damping:    0
    Advertised prefixes:          4
    Last traffic (seconds): Received 19    Sent 26    Checked 9
    Input messages:  Total 256    Updates 3      Refreshes 0    Octets 4931
    Output messages: Total 256    Updates 2      Refreshes 0    Octets 4999
    Output Queue[0]: 0

```

**Meaning** The Local AS: 250 and Local System AS: 200 output shows that Device R2 has the expected settings. Additionally, the output shows that the options list includes LocalAS.

### *Checking the BGP Peering Sessions*

**Purpose** Ensure that the sessions are established and that the local AS number 250 is displayed.

**Action** From operational mode, enter the **show bgp summary** command.

```

user@R1> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History  Damp State   Pending
inet.0          4          2          0          0          0      0        0
Peer           AS        InPkt    OutPkt    OutQ   Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.2        250        232      233        0        4    1:42:37
2/4/4/0        0/0/0/0

```

```

user@R3> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History  Damp State   Pending
inet.0          4          2          0          0          0      0        0
Peer           AS        InPkt    OutPkt    OutQ   Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.1.0.1        250        235      236        0        4    1:44:25
2/4/4/0        0/0/0/0

```

**Meaning** Device R1 and Device R3 appear to be peering with a device in AS 250, even though Device R2 is actually in AS 200.

### *Verifying the BGP AS Paths*

**Purpose** Make sure that the routes are in the routing tables and that the AS paths show the local AS number 250.

**Action** From configuration mode, enter the **set route protocol bgp** command.

```

user@R1> show route protocol bgp
inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30      [BGP/170] 01:46:44, localpref 100
                  AS path: 250 I
                  > to 10.0.0.2 via fe-1/2/0.1
10.1.0.0/30      [BGP/170] 01:46:44, localpref 100
                  AS path: 250 I
                  > to 10.0.0.2 via fe-1/2/0.1
192.168.0.2/32   *[BGP/170] 01:46:44, localpref 100
                  AS path: 250 I
                  > to 10.0.0.2 via fe-1/2/0.1

```

```

192.168.0.3/32      *[BGP/170] 01:46:40, localpref 100
                   AS path: 250 300 I
                   > to 10.0.0.2 via fe-1/2/0.1

user@R3> show route protocol bgp

inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        [BGP/170] 01:47:10, localpref 100
                   AS path: 250 I
                   > to 10.1.0.1 via fe-1/2/0.4
10.1.0.0/30        [BGP/170] 01:47:10, localpref 100
                   AS path: 250 I
                   > to 10.1.0.1 via fe-1/2/0.4
192.168.0.1/32     *[BGP/170] 01:47:10, localpref 100
                   AS path: 250 100 I
                   > to 10.1.0.1 via fe-1/2/0.4
192.168.0.2/32     *[BGP/170] 01:47:10, localpref 100
                   AS path: 250 I
                   > to 10.1.0.1 via fe-1/2/0.4

```

**Meaning** The output shows that Device R1 and Device R3 appear to have routes with AS paths that include AS 250, even though Device R2 is actually in AS 200.

## Example: Configuring a Private Local AS for EBGp Sessions

This example shows how to configure a private local autonomous system (AS) number. The local AS is considered to be private because it is advertised to peers that use the local AS number for peering, but is hidden in the announcements to peers that can use the global AS number for peering.

- [Requirements on page 146](#)
- [Overview on page 146](#)
- [Configuration on page 148](#)
- [Verification on page 150](#)

### Requirements

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

### Overview

Use the **local-as** statement when ISPs merge and want to preserve a customer's configuration, particularly the AS with which the customer is configured to establish a peer relationship. The **local-as** statement simulates the AS number already in place in customer routers, even if the ISP's router has moved to a different AS.

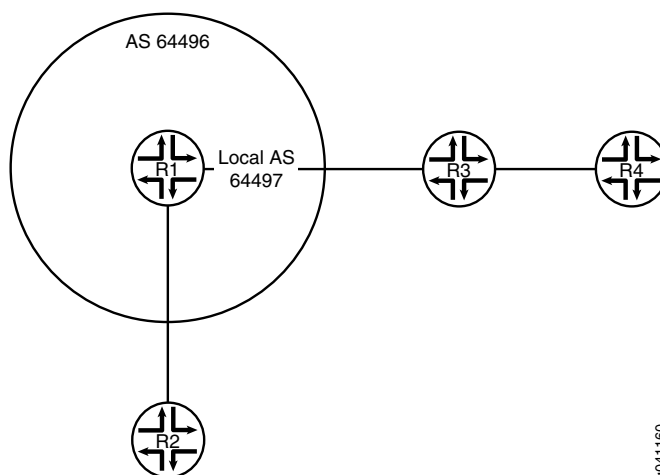
When you use the **private** option, the local AS is used during the establishment of the BGP session with an external BGP (EBGP) neighbor, but is hidden in the AS path sent to other EBGp peers. Only the global AS is included in the AS path sent to external peers.

The **private** option is useful for establishing local peering with routing devices that remain configured with their former AS or with a specific customer that has not yet modified its peer arrangements. The local AS is used to establish the BGP session with the EBGP neighbor, but is hidden in the AS path sent to external peers in another AS.

Include the **private** option so that the local AS is not prepended before the global AS in the AS path sent to external peers. When you specify the **private** option, the local AS is prepended only in the AS path sent to the EBGP neighbor.

Figure 22 on page 147 shows the sample topology.

**Figure 22: Topology for Configuring a Private Local AS**



Device R1 is in AS 64496. Device R2 is in AS 64510. Device R3 is in AS 64511. Device R4 is in AS 64512. Device R1 formerly belonged to AS 64497, which has merged with another network and now belongs to AS 64496. Because Device R3 still peers with Device R1, using its former AS, 64497, Device R1 needs to be configured with a local AS of 64497 in order to maintain peering with Device R3. Configuring a local AS of 64497 permits Device R1 to add AS 64497 when advertising routes to Device R3. Device R3 sees an AS path of 64497 64496 for the prefix 10.1.1.2/32, which is Device R2's loopback interface. Device R4, which is behind Device R3, sees an AS path of 64511 64497 64496 64510 to Device R2's loopback interface. To prevent Device R1 from adding the local AS number in its announcements to other peers, this example includes the **local-as 64497 private** statement. The **private** option configures Device R1 to not include the local AS 64497 when announcing routes to other peers. Device R2 sees an AS path of 64496 64511 to Device R3 and an AS path of 64496 64511 64512 to Device R4. The **private** option in Device R1's configuration causes the AS number 64497 to be missing from the AS paths that Device R1 readvertises to Device R2.

Device R2 is hiding the private local AS from all the routers, except Device R3. The **private** option applies to the routes that Device R1 receives (learns) from Device R3 and that Device R1, in turn, readvertises to other routers. When these routes learned from Device R3 are readvertised by Device R1 to Device R2, the private local AS is missing from the AS path advertised to Device R2.

## Configuration

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CLI Quick Configuration</b> | 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 <b>[edit]</b> hierarchy level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Device R1</b>               | <pre> set interfaces fe-1/2/0 unit 3 family inet address 192.168.1.1/24 set interfaces fe-1/2/1 unit 5 family inet address 192.168.10.1/24 set interfaces lo0 unit 2 family inet address 10.1.1.1/32 set protocols bgp group external-AS64511 type external set protocols bgp group external-AS64511 peer-as 64511 set protocols bgp group external-AS64511 local-as 64497 set protocols bgp group external-AS64511 local-as private set protocols bgp group external-AS64511 neighbor 192.168.1.2 set protocols bgp group external-AS64510 type external set protocols bgp group external-AS64510 peer-as 64510 set protocols bgp group external-AS64510 neighbor 192.168.10.2 set policy-options policy-statement send-direct term 1 from protocol direct set policy-options policy-statement send-direct term 1 then accept set routing-options autonomous-system 64496 </pre> |
| <b>Device R2</b>               | <pre> set interfaces fe-1/2/0 unit 6 family inet address 192.168.10.2/24 set interfaces lo0 unit 3 family inet address 10.1.1.2/32 set protocols bgp group external type external set protocols bgp group external export send-direct set protocols bgp group external peer-as 64496 set protocols bgp group external neighbor 192.168.10.1 set policy-options policy-statement send-direct term 1 from protocol direct set policy-options policy-statement send-direct term 1 then accept set routing-options autonomous-system 64510 </pre>                                                                                                                                                                                                                                                                                                                                     |
| <b>Device R3</b>               | <pre> set interfaces fe-1/2/0 unit 4 family inet address 192.168.1.2/24 set interfaces fe-1/2/1 unit 7 family inet address 192.168.5.1/24 set interfaces lo0 unit 4 family inet address 10.1.1.3/32 set protocols bgp group external type external set protocols bgp group external export send-direct set protocols bgp group external neighbor 192.168.1.1 peer-as 64497 set protocols bgp group external neighbor 192.168.5.2 peer-as 64512 set policy-options policy-statement send-direct term 1 from protocol direct set policy-options policy-statement send-direct term 1 then accept set routing-options autonomous-system 64511 </pre>                                                                                                                                                                                                                                  |
| <b>Device R4</b>               | <pre> set interfaces fe-1/2/0 unit 8 family inet address 192.168.5.2/24 set interfaces lo0 unit 5 family inet address 10.1.1.4/32 set protocols bgp group external type external set protocols bgp group external export send-direct set protocols bgp group external peer-as 64511 set protocols bgp group external neighbor 192.168.5.1 set policy-options policy-statement send-direct term 1 from protocol direct set policy-options policy-statement send-direct term 1 then accept set routing-options autonomous-system 64512 </pre>                                                                                                                                                                                                                                                                                                                                       |

**Configuring Device R1**

**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 Device R1:

1. Configure the interfaces.
 

```
[edit interfaces fe-1/2/0 unit 3]
user@R1# set family inet address 192.168.1.1/24

[edit interfaces fe-1/2/1 unit 5]
user@R1# set family inet address 192.168.10.1/24

[edit interfaces lo0 unit 2]
user@R1# set family inet address 10.1.1.1/32
```
2. Configure the EBGP peering session with Device R2.
 

```
[edit protocols bgp group external-AS64510]
user@R1# set type external
user@R1# set peer-as 64510
user@R1# set neighbor 192.168.10.2
```
3. Configure the EBGP peering session with Device R3.
 

```
[edit protocols bgp group external-AS64511]
user@R1# set type external
user@R1# set peer-as 64511
user@R1# set local-as 64497
user@R1# set local-as private
user@R1# set neighbor 192.168.1.2
```
4. Configure the routing policy.
 

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```
5. Configure the global autonomous system (AS) number.
 

```
[edit routing-options]
user@R1# set autonomous-system 64496
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 192.168.1.1/24;
    }
  }
}
```

```
    }
  }
  fe-1/2/1 {
    unit 5 {
      family inet {
        address 192.168.10.1/24;
      }
    }
  }
  lo0 {
    unit 2 {
      family inet {
        address 10.1.1.1/32;
      }
    }
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show protocols
bgp {
  group external-AS64511 {
    type external;
    peer-as 64511;
    local-as 64497 private;
    neighbor 192.168.1.2;
  }
  group external-AS64510 {
    type external;
    peer-as 64510;
    neighbor 192.168.10.2;
  }
}

user@R1# show routing-options
autonomous-system 64496;
```

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

Repeat the configuration as needed for the other devices in the topology.

---

### Verification

Confirm that the configuration is working properly.

- [Checking Device R2's AS Paths on page 151](#)
- [Checking Device R3's AS Paths on page 151](#)

**Checking Device R2's AS Paths**

**Purpose** Make sure that Device R2 does not have AS 64497 in its AS paths to Device R3 and Device R4.

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

```
user@R2> show route protocol bgp
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.3/32      *[BGP/170] 01:33:11, localpref 100
                  AS path: 64496 64511 I
                  > to 192.168.10.1 via fe-1/2/0.6
10.1.1.4/32      *[BGP/170] 01:33:11, localpref 100
                  AS path: 64496 64511 64512 I
                  > to 192.168.10.1 via fe-1/2/0.6
192.168.5.0/24   *[BGP/170] 01:49:15, localpref 100
                  AS path: 64496 64511 I
                  > to 192.168.10.1 via fe-1/2/0.6
```

**Meaning** Device R2's AS paths do not include AS 64497.

**Checking Device R3's AS Paths**

**Purpose** Make sure that Device R3 does not have AS 64497 in its AS path to Device R4.

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

```
user@R3> show route protocol bgp
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.2/32      *[BGP/170] 01:35:11, localpref 100
                  AS path: 64497 64496 64510 I
                  > to 192.168.1.1 via fe-1/2/0.4
10.1.1.4/32      *[BGP/170] 01:35:11, localpref 100
                  AS path: 64512 I
                  > to 192.168.5.2 via fe-1/2/1.7
192.168.5.0/24   [BGP/170] 01:51:15, localpref 100
                  AS path: 64512 I
                  > to 192.168.5.2 via fe-1/2/1.7
```

**Meaning** Device R3's route to Device R2 (prefix 10.1.1.2) includes both the local and the global AS configured on Device R1 (64497 and 64496, respectively).

**Related Documentation**

- [Examples: Configuring External BGP Peering on page 33](#)
- [BGP Configuration Overview on page 12](#)

**Example: Configuring the Accumulated IGP Attribute for BGP**

- [Understanding the Accumulated IGP Attribute for BGP on page 152](#)
- [Example: Configuring the Accumulated IGP Attribute for BGP on page 152](#)

## Understanding the Accumulated IGP Attribute for BGP

The interior gateway protocols (IGPs) are designed to handle routing within a single domain or an autonomous system (AS). Each link is assigned a particular value called a metric. The distance between the two nodes is calculated as a sum of all the metric values of links along the path. The IGP selects the shortest path between two nodes based on distance.

BGP is designed to provide routing over a large number of independent ASs with limited or no coordination among respective administrations. BGP does not use metrics in the path selection decisions.

The accumulated IGP (AIGP) metric attribute for BGP enables deployment in which a single administration can run several contiguous BGP ASs. Such deployments allow BGP to make routing decisions based on the IGP metric. In such networks, it is possible for BGP to select paths based on metrics as is done by IGPs. In this case, BGP chooses the shortest path between two nodes, even though the nodes might be in two different ASs.

The AIGP attribute is particularly useful in networks that use tunneling to deliver a packet to its BGP next hop. The Juniper Networks® Junos® operating system (Junos OS) currently supports the AIGP attribute for two BGP address families, **family inet labeled-unicast** and **family inet6 labeled-unicast**.

AIGP impacts the BGP best-route decision process. The AIGP attribute preference rule is applied after the local-preference rule. The AIGP distance is compared to break a tie. The BGP best-route decision process also impacts the way the interior cost rule is applied if the resolving next hop has an AIGP attribute. Without AIGP enabled, the interior cost of a route is based on the calculation of the metric to the next hop for the route. With AIGP enabled, the resolving AIGP distance is added to the interior cost.

The AIGP attribute is an optional non-transitive BGP path attribute and is specified in Internet draft draft-ietf-idr-aigp-06, *The Accumulated IGP Metric Attribute for BGP*.

## Example: Configuring the Accumulated IGP Attribute for BGP

This example shows how to configure the accumulated IGP (AIGP) metric attribute for BGP.

- [Requirements on page 152](#)
- [Overview on page 153](#)
- [Configuration on page 154](#)
- [Verification on page 184](#)

---

### Requirements

This example uses the following hardware and software components:

- Seven BGP-speaking devices.
- Junos OS Release 12.1 or later.

## Overview

The AIGP attribute enables deployments in which a single administration can run several contiguous BGP autonomous systems (ASs). Such deployments allow BGP to make routing decisions based on the IGP metric. With AIGP enabled, BGP can select paths based on IGP metrics. This enables BGP to choose the shortest path between two nodes, even though the nodes might be in different ASs. The AIGP attribute is particularly useful in networks that use tunneling to deliver a packet to its BGP next hop. This example shows AIGP configured with MPLS label-switched paths.

To enable AIGP, you include the **aigp** statement in the BGP configuration on a protocol family basis. Configuring AIGP on a particular family enables sending and receiving of the AIGP attribute on that family. By default, AIGP is disabled. An AIGP-disabled neighbor does not send an AIGP attribute and silently discards a received AIGP attribute.

Junos OS supports AIGP for **family inet labeled-unicast** and **family inet6 labeled-unicast**. The **aigp** statement can be configured for a given family at the global BGP, group, or neighbor level.

By default, the value of the AIGP attribute for a local prefix is zero. An AIGP-enabled neighbor can originate an AIGP attribute for a given prefix by export policy, using the **aigp-originate** policy action. The value of the AIGP attribute reflects the IGP distance to the prefix. Alternatively, you can specify a value, by using the **aigp-originate distance distance** policy action. The configurable range is 0 through 4,294,967,295. Only one node needs to originate an AIGP attribute. The AIGP attribute is retained and readvertised if the neighbors are AIGP enabled with the **aigp** statement in the BGP configuration.

The policy action to originate the AIGP attribute has the following requirements:

- Neighbor must be AIGP enabled.
- Policy must be applied as an export policy.
- Prefix must have no current AIGP attribute.
- Prefix must export with next-hop self.
- Prefix must reside within the AIGP domain. Typically, a loopback IP address is the prefix to originate.

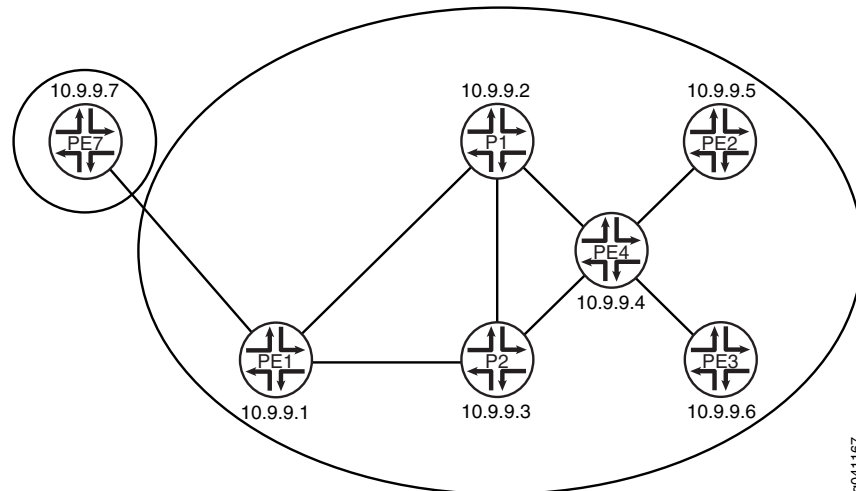
The policy is ignored if these requirements are not met.

## Topology Diagram

Figure 23 on page 154 shows the topology used in this example. OSPF is used as the interior gateway protocol (IGP). Internal BGP (IBGP) is configured between Device PE1 and Device PE4. External BGP (EBGP) is configured between Device PE7 and Device PE1, between Device PE4 and Device PE3, and between Device PE4 and Device PE2. Devices PE4, PE2, and PE3 are configured for multihop. Device PE4 selects a path based on the AIGP value and then readvertises the AIGP value based on the AIGP and policy configuration. Device PE1 readvertises the AIGP value to Device PE7, which is in another administrative domain. Every device has two loopback interface addresses: 10.9.9.x is used for BGP peering and the router ID, and 10.100.1.x is used for the BGP next hop.

The network between Device PE1 and PE3 has IBGP peering and multiple OSPF areas. The external link to Device PE7 is configured to show that the AIGP attribute is readadvertised to a neighbor outside of the administrative domain, if that neighbor is AIGP enabled.

**Figure 23: Advertisement of Multiple Paths in BGP**



For origination of an AIGP attribute, the BGP next hop is required to be itself. If the BGP next hop remains unchanged, the received AIGP attribute is readadvertised, as is, to another AIGP neighbor. If the next hop changes, the received AIGP attribute is readadvertised with an increased value to another AIGP neighbor. The increase in value reflects the IGP distance to the previous BGP next hop. To demonstrate, this example uses loopback interface addresses for Device PE4's EBGP peering sessions with Device PE2 and Device PE3. Multihop is enabled on these sessions so that a recursive lookup is performed to determine the point-to-point interface. Because the next hop changes, the IGP distance is added to the AIGP distance.

### Configuration

- [Configuring Device P1 on page 160](#)
- [Configuring Device P2 on page 163](#)
- [Configuring Device PE4 on page 166](#)
- [Configuring Device PE1 on page 171](#)
- [Configuring Device PE2 on page 175](#)
- [Configuring Device PE3 on page 179](#)
- [Configuring Device PE7 on page 182](#)

#### 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 P1 set interfaces fe-1/2/0 unit 1 description P1-to-PE1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.2/30
set interfaces fe-1/2/0 unit 1 family mpls
```

```

set interfaces fe-1/2/1 unit 4 description P1-to-P2
set interfaces fe-1/2/1 unit 4 family inet address 10.0.0.29/30
set interfaces fe-1/2/1 unit 4 family mpls
set interfaces fe-1/2/2 unit 8 description P1-to-PE4
set interfaces fe-1/2/2 unit 8 family inet address 10.0.0.17/30
set interfaces fe-1/2/2 unit 8 family mpls
set interfaces lo0 unit 3 family inet address 10.9.9.2/32
set interfaces lo0 unit 3 family inet address 10.100.1.2/32
set protocols rsvp interface fe-1/2/0.1
set protocols rsvp interface fe-1/2/2.8
set protocols rsvp interface fe-1/2/1.4
set protocols mpls label-switched-path P1-to-P2 to 10.9.9.3
set protocols mpls label-switched-path P1-to-PE1 to 10.9.9.1
set protocols mpls label-switched-path P1-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.1
set protocols mpls interface fe-1/2/2.8
set protocols mpls interface fe-1/2/1.4
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.2
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group internal neighbor 10.9.9.4
set protocols ospf area 0.0.0.1 interface fe-1/2/0.1 metric 1
set protocols ospf area 0.0.0.1 interface fe-1/2/1.4 metric 1
set protocols ospf area 0.0.0.0 interface fe-1/2/2.8 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.2 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.2 metric 1
set protocols ospf area 0.0.0.0 interface 10.100.1.2 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.2 metric 1
set routing-options router-id 10.9.9.2
set routing-options autonomous-system 13979

```

#### Device P2

```

set interfaces fe-1/2/0 unit 3 description P2-to-PE1
set interfaces fe-1/2/0 unit 3 family inet address 10.0.0.6/30
set interfaces fe-1/2/0 unit 3 family mpls
set interfaces fe-1/2/1 unit 5 description P2-to-P1
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.30/30
set interfaces fe-1/2/1 unit 5 family mpls
set interfaces fe-1/2/2 unit 6 description P2-to-PE4
set interfaces fe-1/2/2 unit 6 family inet address 10.0.0.13/30
set interfaces fe-1/2/2 unit 6 family mpls
set interfaces lo0 unit 5 family inet address 10.9.9.3/32
set interfaces lo0 unit 5 family inet address 10.100.1.3/32
set protocols rsvp interface fe-1/2/1.5
set protocols rsvp interface fe-1/2/2.6
set protocols rsvp interface fe-1/2/0.3
set protocols mpls label-switched-path P2-to-PE1 to 10.9.9.1
set protocols mpls label-switched-path P2-to-P1 to 10.9.9.2
set protocols mpls label-switched-path P2-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/1.5
set protocols mpls interface fe-1/2/2.6
set protocols mpls interface fe-1/2/0.3
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.3
set protocols bgp group internal family inet labeled-unicast aigp

```

```

set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group internal neighbor 10.9.9.4
set protocols ospf area 0.0.0.0 interface fe-1/2/2.6 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.3 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.3 metric 1
set protocols ospf area 0.0.0.0 interface 10.100.1.3 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.3 metric 1
set routing-options router-id 10.9.9.3
set routing-options autonomous-system 13979

```

**Device PE4**

```

set interfaces fe-1/2/0 unit 7 description PE4-to-P2
set interfaces fe-1/2/0 unit 7 family inet address 10.0.0.14/30
set interfaces fe-1/2/0 unit 7 family mpls
set interfaces fe-1/2/1 unit 9 description PE4-to-P1
set interfaces fe-1/2/1 unit 9 family inet address 10.0.0.18/30
set interfaces fe-1/2/1 unit 9 family mpls
set interfaces fe-1/2/2 unit 10 description PE4-to-PE2
set interfaces fe-1/2/2 unit 10 family inet address 10.0.0.21/30
set interfaces fe-1/2/2 unit 10 family mpls
set interfaces fe-1/0/2 unit 12 description PE4-to-PE3
set interfaces fe-1/0/2 unit 12 family inet address 10.0.0.25/30
set interfaces fe-1/0/2 unit 12 family mpls
set interfaces lo0 unit 7 family inet address 10.9.9.4/32
set interfaces lo0 unit 7 family inet address 10.100.1.4/32
set protocols rsvp interface fe-1/2/0.7
set protocols rsvp interface fe-1/2/1.9
set protocols rsvp interface fe-1/2/2.10
set protocols rsvp interface fe-1/0/2.12
set protocols mpls label-switched-path PE4-to-PE2 to 10.9.9.5
set protocols mpls label-switched-path PE4-to-PE3 to 10.9.9.6
set protocols mpls label-switched-path PE4-to-P1 to 10.9.9.2
set protocols mpls label-switched-path PE4-to-P2 to 10.9.9.3
set protocols mpls interface fe-1/2/0.7
set protocols mpls interface fe-1/2/1.9
set protocols mpls interface fe-1/2/2.10
set protocols mpls interface fe-1/0/2.12
set protocols bgp export next-hop
set protocols bgp export aigp
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.4
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.4
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external peer-as 7018
set protocols bgp group external neighbor 10.9.9.5
set protocols bgp group external neighbor 10.9.9.6
set protocols ospf area 0.0.0.0 interface fe-1/2/1.9 metric 1
set protocols ospf area 0.0.0.0 interface fe-1/2/0.7 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.4 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.4 metric 1

```

```

set protocols ospf area 0.0.0.0 interface 10.100.1.4 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.4 metric 1
set protocols ospf area 0.0.0.2 interface fe-1/2/2.10 metric 1
set protocols ospf area 0.0.0.3 interface fe-1/0/2.12 metric 1
set policy-options policy-statement aigp term 10 from protocol static
set policy-options policy-statement aigp term 10 from route-filter 44.0.0.0/24 exact
set policy-options policy-statement aigp term 10 then aigp-originate distance 200
set policy-options policy-statement aigp term 10 then next-hop 10.100.1.4
set policy-options policy-statement aigp term 10 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.4
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.4/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.4/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.4
set policy-options policy-statement next-hop term 20 then accept
set routing-options static route 44.0.0.0/24 discard
set routing-options router-id 10.9.9.4
set routing-options autonomous-system 13979

```

**Device PE1**

```

set interfaces fe-1/2/0 unit 0 description PE1-to-P1
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 2 description PE1-to-P2
set interfaces fe-1/2/1 unit 2 family inet address 10.0.0.5/30
set interfaces fe-1/2/1 unit 2 family mpls
set interfaces fe-1/2/2 unit 14 description PE1-to-PE7
set interfaces fe-1/2/2 unit 14 family inet address 10.0.0.9/30
set interfaces lo0 unit 1 family inet address 10.9.9.1/32
set interfaces lo0 unit 1 family inet address 10.100.1.1/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface fe-1/2/1.2
set protocols rsvp interface fe-1/2/2.14
set protocols mpls label-switched-path PE1-to-P1 to 10.9.9.2
set protocols mpls label-switched-path PE1-to-P2 to 10.9.9.3
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/1.2
set protocols mpls interface fe-1/2/2.14
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.1
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal export SET_EXPORT_ROUTES
set protocols bgp group internal vpn-apply-export
set protocols bgp group internal neighbor 10.9.9.4
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group external type external
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external peer-as 7019
set protocols bgp group external neighbor 10.0.0.10
set protocols ospf area 0.0.0.1 interface fe-1/2/0.0 metric 1
set protocols ospf area 0.0.0.1 interface fe-1/2/1.2 metric 1
set protocols ospf area 0.0.0.1 interface 10.9.9.1 passive

```

```

set protocols ospf area 0.0.0.1 interface 10.9.9.1 metric 1
set protocols ospf area 0.0.0.1 interface 10.100.1.1 passive
set protocols ospf area 0.0.0.1 interface 10.100.1.1 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set routing-options router-id 10.9.9.1
set routing-options autonomous-system 13979

```

**Device PE2**

```

set interfaces fe-1/2/0 unit 11 description PE2-to-PE4
set interfaces fe-1/2/0 unit 11 family inet address 10.0.0.22/30
set interfaces fe-1/2/0 unit 11 family mpls
set interfaces lo0 unit 9 family inet address 10.9.9.5/32 primary
set interfaces lo0 unit 9 family inet address 10.100.1.5/32
set protocols rsvp interface fe-1/2/0.11
set protocols mpls label-switched-path PE2-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.11
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.5
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export next-hop
set protocols bgp group external export aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external vpn-apply-export
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.9.9.4
set protocols ospf area 0.0.0.2 interface 10.9.9.5 passive
set protocols ospf area 0.0.0.2 interface 10.9.9.5 metric 1
set protocols ospf area 0.0.0.2 interface 10.100.1.5 passive
set protocols ospf area 0.0.0.2 interface 10.100.1.5 metric 1
set protocols ospf area 0.0.0.2 interface fe-1/2/0.11 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol static
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.5
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set policy-options policy-statement aigp term 10 from route-filter 55.0.0.0/24 exact
set policy-options policy-statement aigp term 10 then aigp-originate distance 20
set policy-options policy-statement aigp term 10 then next-hop 10.100.1.5
set policy-options policy-statement aigp term 10 then accept
set policy-options policy-statement aigp term 20 from route-filter 99.0.0.0/24 exact
set policy-options policy-statement aigp term 20 then aigp-originate distance 30
set policy-options policy-statement aigp term 20 then next-hop 10.100.1.5
set policy-options policy-statement aigp term 20 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.5
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.5/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.5/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.5

```

```

set policy-options policy-statement next-hop term 20 then accept
set routing-options static route 99.0.0.0/24 discard
set routing-options static route 55.0.0.0/24 discard
set routing-options router-id 10.9.9.5
set routing-options autonomous-system 7018

```

**Device PE3**

```

set interfaces fe-1/2/0 unit 13 description PE3-to-PE4
set interfaces fe-1/2/0 unit 13 family inet address 10.0.0.26/30
set interfaces fe-1/2/0 unit 13 family mpls
set interfaces lo0 unit 11 family inet address 10.9.9.6/32
set interfaces lo0 unit 11 family inet address 10.100.1.6/32
set protocols rsvp interface fe-1/2/0.13
set protocols mpls label-switched-path PE3-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.13
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.6
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export next-hop
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external vpn-apply-export
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.9.9.4
set protocols ospf area 0.0.0.3 interface 10.9.9.6 passive
set protocols ospf area 0.0.0.3 interface 10.9.9.6 metric 1
set protocols ospf area 0.0.0.3 interface 10.100.1.6 passive
set protocols ospf area 0.0.0.3 interface 10.100.1.6 metric 1
set protocols ospf area 0.0.0.3 interface fe-1/2/0.13 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol static
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.6
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.6
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.6/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.6/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.6
set policy-options policy-statement next-hop term 20 then accept
set routing-options router-id 10.9.9.6
set routing-options autonomous-system 7018

```

**Device PE7**

```

set interfaces fe-1/2/0 unit 15 description PE7-to-PE1
set interfaces fe-1/2/0 unit 15 family inet address 10.0.0.10/30
set interfaces lo0 unit 13 family inet address 10.9.9.7/32
set interfaces lo0 unit 13 family inet address 10.100.1.7/32
set protocols bgp group external type external
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.0.0.9

```

```

set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.7
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set routing-options router-id 10.9.9.7
set routing-options autonomous-system 7019

```

### Configuring Device P1

**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 Device P1:

1. Configure the interfaces.

```

[edit interfaces]
user@P1# set fe-1/2/0 unit 1 description P1-to-PE1
user@P1# set fe-1/2/0 unit 1 family inet address 10.0.0.2/30
user@P1# set fe-1/2/0 unit 1 family mpls
user@P1# set fe-1/2/1 unit 4 description P1-to-P2
user@P1# set fe-1/2/1 unit 4 family inet address 10.0.0.29/30
user@P1# set fe-1/2/1 unit 4 family mpls
user@P1# set fe-1/2/2 unit 8 description P1-to-PE4
user@P1# set fe-1/2/2 unit 8 family inet address 10.0.0.17/30
user@P1# set fe-1/2/2 unit 8 family mpls
user@P1# set lo0 unit 3 family inet address 10.9.9.2/32
user@P1# set lo0 unit 3 family inet address 10.100.1.2/32

```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```

[edit protocols]
user@P1# set rsvp interface fe-1/2/0.1
user@P1# set rsvp interface fe-1/2/2.8
user@P1# set rsvp interface fe-1/2/1.4
user@P1# set mpls label-switched-path P1-to-P2 to 10.9.9.3
user@P1# set mpls label-switched-path P1-to-PE1 to 10.9.9.1
user@P1# set mpls label-switched-path P1-to-PE4 to 10.9.9.4
user@P1# set mpls interface fe-1/2/0.1
user@P1# set mpls interface fe-1/2/2.8
user@P1# set mpls interface fe-1/2/1.4

```

3. Configure BGP.

```

[edit protocols bgp group internal]
user@P1# set type internal
user@P1# set local-address 10.9.9.2
user@P1# set neighbor 10.9.9.1
user@P1# set neighbor 10.9.9.3
user@P1# set neighbor 10.9.9.4

```

4. Enable AIGP.

```

[edit protocols bgp group internal]
user@P1# set family inet labeled-unicast aigp

```

5. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf]
user@P1# set area 0.0.0.1 interface fe-1/2/0.1 metric 1
user@P1# set area 0.0.0.1 interface fe-1/2/1.4 metric 1
user@P1# set area 0.0.0.0 interface fe-1/2/2.8 metric 1
user@P1# set area 0.0.0.0 interface 10.9.9.2 passive
user@P1# set area 0.0.0.0 interface 10.9.9.2 metric 1
user@P1# set area 0.0.0.0 interface 10.100.1.2 passive
user@P1# set area 0.0.0.0 interface 10.100.1.2 metric 1
```

6. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@P1# set router-id 10.9.9.2
user@P1# set autonomous-system 13979
```

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

```
user@P1# commit
```

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

```
user@P1# show interfaces
fe-1/2/0 {
  unit 1 {
    description P1-to-PE1;
    family inet {
      address 10.0.0.2/30;
    }
    family mpls;
  }
}
fe-1/2/1 {
  unit 4 {
    description P1-to-P2;
    family inet {
      address 10.0.0.29/30;
    }
    family mpls;
  }
}
fe-1/2/2 {
  unit 8 {
    description P1-to-PE4;
    family inet {
      address 10.0.0.17/30;
    }
    family mpls;
  }
}
lo0 {
  unit 3 {
    family inet {
```

```
        address 10.9.9.2/32;
        address 10.100.1.2/32;
    }
}

user@P1# show protocols
rsvp {
    interface fe-1/2/0.1;
    interface fe-1/2/2.8;
    interface fe-1/2/1.4;
}
mpls {
    label-switched-path P1-to-P2 {
        to 10.9.9.3;
    }
    label-switched-path P1-to-PE1 {
        to 10.9.9.1;
    }
    label-switched-path P1-to-PE4 {
        to 10.9.9.4;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/2.8;
    interface fe-1/2/1.4;
}
bgp {
    group internal {
        type internal;
        local-address 10.9.9.2;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        neighbor 10.9.9.1;
        neighbor 10.9.9.3;
        neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.1 {
        interface fe-1/2/0.1 {
            metric 1;
        }
        interface fe-1/2/1.4 {
            metric 1;
        }
    }
    area 0.0.0.0 {
        interface fe-1/2/2.8 {
            metric 1;
        }
        interface 10.9.9.2 {
            passive;
            metric 1;
        }
    }
}
```

```

    }
    interface 10.100.1.2 {
        passive;
        metric 1;
    }
}

user@P1# show routing-options
router-id 10.9.9.2;
autonomous-system 13979;

```

### Configuring Device P2

**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 Device P2:

1. Configure the interfaces.

```

[edit interfaces]
user@P2# set fe-1/2/0 unit 3 description P2-to-PE1
user@P2# set fe-1/2/0 unit 3 family inet address 10.0.0.6/30
user@P2# set fe-1/2/0 unit 3 family mpls
user@P2# set fe-1/2/1 unit 5 description P2-to-P1
user@P2# set fe-1/2/1 unit 5 family inet address 10.0.0.30/30
user@P2# set fe-1/2/1 unit 5 family mpls
user@P2# set fe-1/2/2 unit 6 description P2-to-PE4
user@P2# set fe-1/2/2 unit 6 family inet address 10.0.0.13/30
user@P2# set fe-1/2/2 unit 6 family mpls
user@P2# set lo0 unit 5 family inet address 10.9.9.3/32
user@P2# set lo0 unit 5 family inet address 10.100.1.3/32

```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```

[edit protocols]
user@P2# set rsvp interface fe-1/2/1.5
user@P2# set rsvp interface fe-1/2/2.6
user@P2# set rsvp interface fe-1/2/0.3
user@P2# set mpls label-switched-path P2-to-PE1 to 10.9.9.1
user@P2# set mpls label-switched-path P2-to-P1 to 10.9.9.2
user@P2# set mpls label-switched-path P2-to-PE4 to 10.9.9.4
user@P2# set mpls interface fe-1/2/1.5
user@P2# set mpls interface fe-1/2/2.6
user@P2# set mpls interface fe-1/2/0.3

```

3. Configure BGP.

```

[edit protocols bgp group internal]
user@P2# set type internal
user@P2# set local-address 10.9.9.3
user@P2# set neighbor 10.9.9.1
user@P2# set neighbor 10.9.9.2
user@P2# set neighbor 10.9.9.4

```

4. Enable AIGP.

```
[edit protocols bgp group internal]
user@P2# set family inet labeled-unicast aigp
```

5. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf]
user@P2# set area 0.0.0.0 interface fe-1/2/2.6 metric 1
user@P2# set area 0.0.0.0 interface 10.9.9.3 passive
user@P2# set area 0.0.0.0 interface 10.9.9.3 metric 1
user@P2# set area 0.0.0.0 interface 10.100.1.3 passive
user@P2# set area 0.0.0.0 interface 10.100.1.3 metric 1
```

6. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@P2# set router-id 10.9.9.3
user@P2# set autonomous-system 13979
```

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

```
user@P2# commit
```

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

```
user@P2# show interfaces
fe-1/2/0 {
  unit 3 {
    description P2-to-PE1;
    family inet {
      address 10.0.0.6/30;
    }
    family mpls;
  }
}
fe-1/2/1 {
  unit 5 {
    description P2-to-P1;
    family inet {
      address 10.0.0.30/30;
    }
    family mpls;
  }
}
fe-1/2/2 {
  unit 6 {
    description P2-to-PE4;
    family inet {
      address 10.0.0.13/30;
    }
    family mpls;
  }
}
lo0 {
```

```

unit 5 {
    family inet {
        address 10.9.9.3/32;
        address 10.100.1.3/32;
    }
}

user@P2# show protocols
rsvp {
    interface fe-1/2/1.5;
    interface fe-1/2/2.6;
    interface fe-1/2/0.3;
}
mpls {
    label-switched-path P2-to-PE1 {
        to 10.9.9.1;
    }
    label-switched-path P2-to-P1 {
        to 10.9.9.2;
    }
    label-switched-path P2-to-PE4 {
        to 10.9.9.4;
    }
    interface fe-1/2/1.5;
    interface fe-1/2/2.6;
    interface fe-1/2/0.3;
}
bgp {
    group internal {
        type internal;
        local-address 10.9.9.3;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        neighbor 10.9.9.1;
        neighbor 10.9.9.2;
        neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/2.6 {
            metric 1;
        }
        interface 10.9.9.3 {
            passive;
            metric 1;
        }
        interface 10.100.1.3 {
            passive;
            metric 1;
        }
    }
}

```

```

}
user@P2# show routing-options
router-id 10.9.9.3;
autonomous-system 13979;

```

### Configuring Device PE4

**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 Device PE4:

1. Configure the interfaces.

```

[edit interfaces]
user@PE4# set fe-1/2/0 unit 7 description PE4-to-P2
user@PE4# set fe-1/2/0 unit 7 family inet address 10.0.0.14/30
user@PE4# set fe-1/2/0 unit 7 family mpls
user@PE4# set fe-1/2/1 unit 9 description PE4-to-P1
user@PE4# set fe-1/2/1 unit 9 family inet address 10.0.0.18/30
user@PE4# set fe-1/2/1 unit 9 family mpls
user@PE4# set fe-1/2/2 unit 10 description PE4-to-PE2
user@PE4# set fe-1/2/2 unit 10 family inet address 10.0.0.21/30
user@PE4# set fe-1/2/2 unit 10 family mpls
user@PE4# set fe-1/0/2 unit 12 description PE4-to-PE3
user@PE4# set fe-1/0/2 unit 12 family inet address 10.0.0.25/30
user@PE4# set fe-1/0/2 unit 12 family mpls
user@PE4# set lo0 unit 7 family inet address 10.9.9.4/32
user@PE4# set lo0 unit 7 family inet address 10.100.1.4/32

```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```

[edit protocols]
user@PE4# set rsvp interface fe-1/2/0.7
user@PE4# set rsvp interface fe-1/2/1.9
user@PE4# set rsvp interface fe-1/2/2.10
user@PE4# set rsvp interface fe-1/0/2.12
user@PE4# set mpls label-switched-path PE4-to-PE2 to 10.9.9.5
user@PE4# set mpls label-switched-path PE4-to-PE3 to 10.9.9.6
user@PE4# set mpls label-switched-path PE4-to-P1 to 10.9.9.2
user@PE4# set mpls label-switched-path PE4-to-P2 to 10.9.9.3
user@PE4# set mpls interface fe-1/2/0.7
user@PE4# set mpls interface fe-1/2/1.9
user@PE4# set mpls interface fe-1/2/2.10
user@PE4# set mpls interface fe-1/0/2.12

```

3. Configure BGP.

```

[edit protocols bgp]
user@PE4# set export next-hop
user@PE4# set export aigp
user@PE4# set group internal type internal
user@PE4# set group internal local-address 10.9.9.4
user@PE4# set group internal neighbor 10.9.9.1
user@PE4# set group internal neighbor 10.9.9.3

```

```

user@PE4# set group internal neighbor 10.9.9.2
user@PE4# set group external type external
user@PE4# set group external multihop ttl 2
user@PE4# set group external local-address 10.9.9.4
user@PE4# set group external peer-as 7018
user@PE4# set group external neighbor 10.9.9.5
user@PE4# set group external neighbor 10.9.9.6

```

4. Enable AIGP.

```

[edit protocols bgp]
user@PE4# set group external family inet labeled-unicast aigp
user@PE4# set group internal family inet labeled-unicast aigp

```

5. Originate a prefix, and configure an AIGP distance.

By default, a prefix is originated using the current IGP distance. Optionally, you can configure a distance for the AIGP attribute, using the **distance** option, as shown here.

```

[edit policy-options policy-statement aigp term 10]
user@PE4# set from protocol static
user@PE4# set from route-filter 44.0.0.0/24 exact
user@PE4# set then aigp-originate distance 200
user@PE4# set then next-hop 10.100.1.4
user@PE4# set then accept

```

6. Enable the policies.

```

[edit policy-options policy-statement next-hop]
user@PE4# set term 10 from protocol bgp
user@PE4# set term 10 then next-hop 10.100.1.4
user@PE4# set term 10 then accept
user@PE4# set term 20 from protocol direct
user@PE4# set term 20 from route-filter 10.9.9.4/32 exact
user@PE4# set term 20 from route-filter 10.100.1.4/32 exact
user@PE4# set term 20 then next-hop 10.100.1.4
user@PE4# set term 20 then accept

```

7. Configure a static route.

```

[edit routing-options]
user@PE4# set static route 44.0.0.0/24 discard

```

8. Configure an IGP, such as OSPF, RIP, or IS-IS.

```

[edit protocols ospf]
user@PE4# set area 0.0.0.0 interface fe-1/2/1.9 metric 1
user@PE4# set area 0.0.0.0 interface fe-1/2/0.7 metric 1
user@PE4# set area 0.0.0.0 interface 10.9.9.4 passive
user@PE4# set area 0.0.0.0 interface 10.9.9.4 metric 1
user@PE4# set area 0.0.0.0 interface 10.100.1.4 passive
user@PE4# set area 0.0.0.0 interface 10.100.1.4 metric 1
user@PE4# set area 0.0.0.2 interface fe-1/2/2.10 metric 1
user@PE4# set area 0.0.0.3 interface fe-1/0/2.12 metric 1

```

9. Configure the router ID and the autonomous system number.

```

[edit routing-options]

```

```
user@PE4# set router-id 10.9.9.4
user@PE4# set autonomous-system 13979
```

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

```
user@PE4# commit
```

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

```
user@PE4# show interfaces
fe-1/0/2 {
  unit 12 {
    description PE4-to-PE3;
    family inet {
      address 10.0.0.25/30;
    }
    family mpls;
  }
}
fe-1/2/0 {
  unit 7 {
    description PE4-to-P2;
    family inet {
      address 10.0.0.14/30;
    }
    family mpls;
  }
}
fe-1/2/1 {
  unit 9 {
    description PE4-to-P1;
    family inet {
      address 10.0.0.18/30;
    }
    family mpls;
  }
}
fe-1/2/2 {
  unit 10 {
    description PE4-to-PE2;
    family inet {
      address 10.0.0.21/30;
    }
    family mpls;
  }
}
lo0 {
  unit 7 {
    family inet {
      address 10.9.9.4/32;
      address 10.100.1.4/32;
    }
  }
}
```

```

    }
  }
user@PE4# show policy-options
policy-statement aigp {
  term 10 {
    from {
      protocol static;
      route-filter 44.0.0.0/24 exact;
    }
    then {
      aigp-originate distance 200;
      next-hop 10.100.1.4;
      accept;
    }
  }
}
policy-statement next-hop {
  term 10 {
    from protocol bgp;
    then {
      next-hop 10.100.1.4;
      accept;
    }
  }
  term 20 {
    from {
      protocol direct;
      route-filter 10.9.9.4/32 exact;
      route-filter 10.100.1.4/32 exact;
    }
    then {
      next-hop 10.100.1.4;
      accept;
    }
  }
}
user@PE4# show protocols
rsvp {
  interface fe-1/2/0.7;
  interface fe-1/2/1.9;
  interface fe-1/2/2.10;
  interface fe-1/0/2.12;
}
mpls {
  label-switched-path PE4-to-PE2 {
    to 10.9.9.5;
  }
  label-switched-path PE4-to-PE3 {
    to 10.9.9.6;
  }
  label-switched-path PE4-to-P1 {
    to 10.9.9.2;
  }
  label-switched-path PE4-to-P2 {
    to 10.9.9.3;
  }
}

```

```
    }
    interface fe-1/2/0.7;
    interface fe-1/2/1.9;
    interface fe-1/2/2.10;
    interface fe-1/0/2.12;
  }
  bgp {
    export [ next-hop aigp ];
    group internal {
      type internal;
      local-address 10.9.9.4;
      family inet {
        labeled-unicast {
          aigp;
        }
      }
      neighbor 10.9.9.1;
      neighbor 10.9.9.3;
      neighbor 10.9.9.2;
    }
    group external {
      type external;
      multihop {
        ttl 2;
      }
      local-address 10.9.9.4;
      family inet {
        labeled-unicast {
          aigp;
        }
      }
      peer-as 7018;
      neighbor 10.9.9.5;
      neighbor 10.9.9.6;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface fe-1/2/1.9 {
        metric 1;
      }
      interface fe-1/2/0.7 {
        metric 1;
      }
      interface 10.9.9.4 {
        passive;
        metric 1;
      }
      interface 10.100.1.4 {
        passive;
        metric 1;
      }
    }
    area 0.0.0.2 {
      interface fe-1/2/2.10 {
        metric 1;
      }
    }
  }
}
```

```

    }
  }
  area 0.0.0.3 {
    interface fe-1/0/2.12 {
      metric 1;
    }
  }
}

user@PE4# show routing-options
static {
  route 44.0.0.0/24 discard;
}
router-id 10.9.9.4;
autonomous-system 13979;

```

### Configuring Device PE1

**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 Device PE1:

1. Configure the interfaces.

```

[edit interfaces]
user@PE1# set fe-1/2/0 unit 0 description PE1-to-P1
user@PE1# set fe-1/2/0 unit 0 family inet address 10.0.0.1/30
user@PE1# set fe-1/2/0 unit 0 family mpls
user@PE1# set fe-1/2/1 unit 2 description PE1-to-P2
user@PE1# set fe-1/2/1 unit 2 family inet address 10.0.0.5/30
user@PE1# set fe-1/2/1 unit 2 family mpls
user@PE1# set fe-1/2/2 unit 14 description PE1-to-PE7
user@PE1# set fe-1/2/2 unit 14 family inet address 10.0.0.9/30
user@PE1# set lo0 unit 1 family inet address 10.9.9.1/32
user@PE1# set lo0 unit 1 family inet address 10.100.1.1/32

```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```

[edit protocols]
user@PE1# set rsvp interface fe-1/2/0.0
user@PE1# set rsvp interface fe-1/2/1.2
user@PE1# set rsvp interface fe-1/2/2.14
user@PE1# set mpls label-switched-path PE1-to-P1 to 10.9.9.2
user@PE1# set mpls label-switched-path PE1-to-P2 to 10.9.9.3
user@PE1# set mpls interface fe-1/2/0.0
user@PE1# set mpls interface fe-1/2/1.2
user@PE1# set mpls interface fe-1/2/2.14

```

3. Configure BGP.

```

[edit protocols bgp]
user@PE1# set group internal type internal
user@PE1# set group internal local-address 10.9.9.1
user@PE1# set group internal export SET_EXPORT_ROUTES
user@PE1# set group internal vpn-apply-export

```

```

user@PE1# set group internal neighbor 10.9.9.4
user@PE1# set group internal neighbor 10.9.9.2
user@PE1# set group internal neighbor 10.9.9.3
user@PE1# set group external type external
user@PE1# set group external export SET_EXPORT_ROUTES
user@PE1# set group external peer-as 7019
user@PE1# set group external neighbor 10.0.0.10

```

4. Enable AIGP.

```

[edit protocols bgp]
user@PE1# set group internal family inet labeled-unicast aigp
user@PE1# set group external family inet labeled-unicast aigp

```

5. Enable the policies.

```

[edit policy-options policy-statement SET_EXPORT_ROUTES term 10]
user@PE1# set from protocol direct
user@PE1# set from protocol bgp
user@PE1# set then next-hop 10.100.1.1
user@PE1# set then accept

```

6. Configure an IGP, such as OSPF, RIP, or IS-IS.

```

[edit protocols ospf area 0.0.0.1]
user@PE1# set interface fe-1/2/0.0 metric 1
user@PE1# set interface fe-1/2/1.2 metric 1
user@PE1# set interface 10.9.9.1 passive
user@PE1# set interface 10.9.9.1 metric 1
user@PE1# set interface 10.100.1.1 passive
user@PE1# set interface 10.100.1.1 metric 1

```

7. Configure the router ID and the autonomous system number.

```

[edit routing-options]
user@PE1# set router-id 10.9.9.1
user@PE1# set autonomous-system 13979

```

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

```

user@PE1# commit

```

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

```

user@PE1# show interfaces
fe-1/2/0 {
  unit 0 {
    description PE1-to-P1;
    family inet {
      address 10.0.0.1/30;
    }
    family mpls;
  }
}
fe-1/2/1 {

```

```

    unit 2 {
      description PE1-to-P2;
      family inet {
        address 10.0.0.5/30;
      }
      family mpls;
    }
  }
  fe-1/2/2 {
    unit 14 {
      description PE1-to-PE7;
      family inet {
        address 10.0.0.9/30;
      }
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.9.9.1/32;
      address 10.100.1.1/32;
    }
  }
}

user@PE1# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct bgp ];
    then {
      next-hop 10.100.1.1;
      accept;
    }
  }
}

user@PE1# show protocols
rsvp {
  interface fe-1/2/0.0;
  interface fe-1/2/1.2;
  interface fe-1/2/2.14;
}
mpls {
  label-switched-path PE1-to-P1 {
    to 10.9.9.2;
  }
  label-switched-path PE1-to-P2 {
    to 10.9.9.3;
  }
  interface fe-1/2/0.0;
  interface fe-1/2/1.2;
  interface fe-1/2/2.14;
}
bgp {
  group internal {
    type internal;
    local-address 10.9.9.1;
  }
}

```

```
family inet {
    labeled-unicast {
        aigp;
    }
}
export SET_EXPORT_ROUTES;
vpn-apply-export;
neighbor 10.9.9.4;
neighbor 10.9.9.2;
neighbor 10.9.9.3;
}
group external {
    type external;
    family inet {
        labeled-unicast {
            aigp;
        }
    }
    export SET_EXPORT_ROUTES;
    peer-as 7019;
    neighbor 10.0.0.10;
}
}
ospf {
    area 0.0.0.1 {
        interface fe-1/2/0.0 {
            metric 1;
        }
        interface fe-1/2/1.2 {
            metric 1;
        }
        interface 10.9.9.1 {
            passive;
            metric 1;
        }
        interface 10.100.1.1 {
            passive;
            metric 1;
        }
    }
}
}
```

```
user@PE1# show routing-options
router-id 10.9.9.1;
autonomous-system 13979;
```

### Configuring Device PE2

**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 Device PE2:

1. Configure the interfaces.

```
[edit interfaces]
user@PE2# set fe-1/2/0 unit 11 description PE2-to-PE4
user@PE2# set fe-1/2/0 unit 11 family inet address 10.0.0.22/30
user@PE2# set fe-1/2/0 unit 11 family mpls
user@PE2# set lo0 unit 9 family inet address 10.9.9.5/32 primary
user@PE2# set lo0 unit 9 family inet address 10.100.1.5/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@PE2# set rsvp interface fe-1/2/0.11
user@PE2# set mpls label-switched-path PE2-to-PE4 to 10.9.9.4
user@PE2# set mpls interface fe-1/2/0.11
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE2# set group external type external
user@PE2# set group external multihop ttl 2
user@PE2# set group external local-address 10.9.9.5
user@PE2# set group external export next-hop
user@PE2# set group external export aigp
user@PE2# set group external export SET_EXPORT_ROUTES
user@PE2# set group external vpn-apply-export
user@PE2# set group external peer-as 13979
user@PE2# set group external neighbor 10.9.9.4
```

4. Enable AIGP.

```
[edit protocols bgp]
user@PE2# set group external family inet labeled-unicast aigp
```

5. Originate a prefix, and configure an AIGP distance.

By default, a prefix is originated using the current IGP distance. Optionally, you can configure a distance for the AIGP attribute, using the **distance** option, as shown here.

```
[edit policy-options policy-statement aigp]
user@PE2# set term 10 from route-filter 55.0.0.0/24 exact
user@PE2# set term 10 then aigp-originate distance 20
user@PE2# set term 10 then next-hop 10.100.1.5
user@PE2# set term 10 then accept
user@PE2# set term 20 from route-filter 99.0.0.0/24 exact
user@PE2# set term 20 then aigp-originate distance 30
user@PE2# set term 20 then next-hop 10.100.1.5
user@PE2# set term 20 then accept
```

6. Enable the policies.

```
[edit policy-options]
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
direct
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
static
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
bgp
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 then next-hop
10.100.1.5
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 then accept
user@PE2# set policy-statement next-hop term 10 from protocol bgp
user@PE2# set policy-statement next-hop term 10 then next-hop 10.100.1.5
user@PE2# set policy-statement next-hop term 10 then accept
user@PE2# set policy-statement next-hop term 20 from protocol direct
user@PE2# set policy-statement next-hop term 20 from route-filter 10.9.9.5/32
exact
user@PE2# set policy-statement next-hop term 20 from route-filter 10.100.1.5/32
exact
user@PE2# set policy-statement next-hop term 20 then next-hop 10.100.1.5
user@PE2# set policy-statement next-hop term 20 then accept
```

7. Enable some static routes.

```
[edit routing-options]
user@PE2# set static route 99.0.0.0/24 discard
user@PE2# set static route 55.0.0.0/24 discard
```

8. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf area 0.0.0.2]
user@PE2# set interface 10.9.9.5 passive
user@PE2# set interface 10.9.9.5 metric 1
user@PE2# set interface 10.100.1.5 passive
user@PE2# set interface 10.100.1.5 metric 1
user@PE2# set interface fe-1/2/0.11 metric 1
```

9. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@PE2# set router-id 10.9.9.5
user@PE2# set autonomous-system 7018
```

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

```
user@PE2# commit
```

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

```
user@PE2# show interfaces
fe-1/2/0 {
  unit 11 {
    description PE2-to-PE4;
```

```

        family inet {
            address 10.0.0.22/30;
        }
        family mpls;
    }
}
lo0 {
    unit 9 {
        family inet {
            address 10.9.9.5/32 {
                primary;
            }
            address 10.100.1.5/32;
        }
    }
}

user@PE2# show policy-options
policy-statement SET_EXPORT_ROUTES {
    term 10 {
        from protocol [ direct static bgp ];
        then {
            next-hop 10.100.1.5;
            accept;
        }
    }
}

policy-statement aigp {
    term 10 {
        from {
            route-filter 55.0.0.0/24 exact;
        }
        then {
            aigp-originate distance 20;
            next-hop 10.100.1.5;
            accept;
        }
    }
    term 20 {
        from {
            route-filter 99.0.0.0/24 exact;
        }
        then {
            aigp-originate distance 30;
            next-hop 10.100.1.5;
            accept;
        }
    }
}

policy-statement next-hop {
    term 10 {
        from protocol bgp;
        then {
            next-hop 10.100.1.5;
            accept;
        }
    }
}

```

```
}
term 20 {
  from {
    protocol direct;
    route-filter 10.9.9.5/32 exact;
    route-filter 10.100.1.5/32 exact;
  }
  then {
    next-hop 10.100.1.5;
    accept;
  }
}
}

user@PE2# show protocols
rsvp {
  interface fe-1/2/0.11;
}
mpls {
  label-switched-path PE2-to-PE4 {
    to 10.9.9.4;
  }
  interface fe-1/2/0.11;
}
bgp {
  group external {
    type external;
    multihop {
      ttl 2;
    }
    local-address 10.9.9.5;
    family inet {
      labeled-unicast {
        aigp;
      }
    }
    export [ next-hop aigp SET_EXPORT_ROUTES ];
    vpn-apply-export;
    peer-as 13979;
    neighbor 10.9.9.4;
  }
}
ospf {
  area 0.0.0.2 {
    interface 10.9.9.5 {
      passive;
      metric 1;
    }
    interface 10.100.1.5 {
      passive;
      metric 1;
    }
    interface fe-1/2/0.11 {
      metric 1;
    }
  }
}
```

```

}
user@PE2# show routing-options
static {
    route 99.0.0.0/24 discard;
    route 55.0.0.0/24 discard;
}
router-id 10.9.9.5;
autonomous-system 7018;

```

### Configuring Device PE3

**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 Device PE3:

1. Configure the interfaces.

```

[edit interfaces]
user@PE3# set fe-1/2/0 unit 13 description PE3-to-PE4
user@PE3# set fe-1/2/0 unit 13 family inet address 10.0.0.26/30
user@PE3# set fe-1/2/0 unit 13 family mpls
user@PE3# set lo0 unit 11 family inet address 10.9.9.6/32
user@PE3# set lo0 unit 11 family inet address 10.100.1.6/32

```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```

[edit protocols]
user@PE3# set rsvp interface fe-1/2/0.13
user@PE3# set mpls label-switched-path PE3-to-PE4 to 10.9.9.4
user@PE3# set mpls interface fe-1/2/0.13

```

3. Configure BGP.

```

[edit protocols bgp group external]
user@PE3# set type external
user@PE3# set multihop ttl 2
user@PE3# set local-address 10.9.9.6
user@PE3# set export next-hop
user@PE3# set export SET_EXPORT_ROUTES
user@PE3# set vpn-apply-export
user@PE3# set peer-as 13979
user@PE3# set neighbor 10.9.9.4

```

4. Enable AIGP.

```

[edit protocols bgp group external]
user@PE3# set family inet labeled-unicast aigp

```

5. Enable the policies.

```

[edit policy-options]
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
direct
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
static

```

```

user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
bgp
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 then next-hop
10.100.1.6
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 then accept
user@PE3# set policy-statement next-hop term 10 from protocol bgp
user@PE3# set policy-statement next-hop term 10 then next-hop 10.100.1.6
user@PE3# set policy-statement next-hop term 10 then accept
user@PE3# set policy-statement next-hop term 20 from protocol direct
user@PE3# set policy-statement next-hop term 20 from route-filter 10.9.9.6/32
exact
user@PE3# set policy-statement next-hop term 20 from route-filter 10.100.1.6/32
exact
user@PE3# set policy-statement next-hop term 20 then next-hop 10.100.1.6
user@PE3# set policy-statement next-hop term 20 then accept

```

6. Configure an IGP, such as OSPF, RIP, or IS-IS.

```

[edit protocols ospf area 0.0.0.3]
user@PE3# set interface 10.9.9.6 passive
user@PE3# set interface 10.9.9.6 metric 1
user@PE3# set interface 10.100.1.6 passive
user@PE3# set interface 10.100.1.6 metric 1
user@PE3# set interface fe-1/2/0.13 metric 1

```

7. Configure the router ID and the autonomous system number.

```

[edit routing-options]
user@PE3# set router-id 10.9.9.6
user@PE3# set autonomous-system 7018

```

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

```

user@PE3# commit

```

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

```

user@PE3# show interfaces
fe-1/2/0 {
  unit 13 {
    description PE3-to-PE4;
    family inet {
      address 10.0.0.26/30;
    }
    family mpls;
  }
}
lo0 {
  unit 11 {
    family inet {
      address 10.9.9.6/32;
      address 10.100.1.6/32;
    }
  }
}

```

```

    }
  }
user@PE3# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct static bgp ];
    then {
      next-hop 10.100.1.6;
      accept;
    }
  }
}
policy-statement next-hop {
  term 10 {
    from protocol bgp;
    then {
      next-hop 10.100.1.6;
      accept;
    }
  }
  term 20 {
    from {
      protocol direct;
      route-filter 10.9.9.6/32 exact;
      route-filter 10.100.1.6/32 exact;
    }
    then {
      next-hop 10.100.1.6;
      accept;
    }
  }
}
user@PE3# show protocols
rsvp {
  interface fe-1/2/0.13;
}
mpls {
  label-switched-path PE3-to-PE4 {
    to 10.9.9.4;
  }
  interface fe-1/2/0.13;
}
bgp {
  group external {
    type external;
    multihop {
      ttl 2;
    }
    local-address 10.9.9.6;
    family inet {
      labeled-unicast {
        aigp;
      }
    }
  }
  export [ next-hop SET_EXPORT_ROUTES ];
}

```

```

        vpn-apply-export;
        peer-as 13979;
        neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.3 {
        interface 10.9.9.6 {
            passive;
            metric 1;
        }
        interface 10.100.1.6 {
            passive;
            metric 1;
        }
        interface fe-1/2/0.13 {
            metric 1;
        }
    }
}

user@PE3# show routing-options
router-id 10.9.9.6;
autonomous-system 7018;

```

### Configuring Device PE7

**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 Device PE7:

1. Configure the interfaces.

```

[edit interfaces]
user@PE7# set fe-1/2/0 unit 15 description PE7-to-PE1
user@PE7# set fe-1/2/0 unit 15 family inet address 10.0.0.10/30
user@PE7# set lo0 unit 13 family inet address 10.9.9.7/32
user@PE7# set lo0 unit 13 family inet address 10.100.1.7/32

```

2. Configure BGP.

```

[edit protocols bgp group external]
user@PE7# set type external
user@PE7# set export SET_EXPORT_ROUTES
user@PE7# set peer-as 13979
user@PE7# set neighbor 10.0.0.9

```

3. Enable AIGP.

```

[edit protocols bgp group external]
user@PE7# set family inet labeled-unicast aigp

```

4. Configure the routing policy.

```

[edit policy-options policy-statement SET_EXPORT_ROUTES term 10]
user@PE7# set from protocol direct

```

```

user@PE7# set from protocol bgp
user@PE7# set then next-hop 10.100.1.7
user@PE7# set then accept

```

5. Configure the router ID and the autonomous system number.

```

[edit routing-options]
user@PE7# set router-id 10.9.9.7
user@PE7# set autonomous-system 7019

```

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

```

user@PE7# commit

```

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

```

user@PE7# show interfaces
interfaces {
  fe-1/2/0 {
    unit 15 {
      description PE7-to-PE1;
      family inet {
        address 10.0.0.10/30;
      }
    }
  }
  lo0 {
    unit 13 {
      family inet {
        address 10.9.9.7/32;
        address 10.100.1.7/32;
      }
    }
  }
}

user@PE7# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct bgp ];
    then {
      next-hop 10.100.1.7;
      accept;
    }
  }
}

user@PE7# show protocols
bgp {
  group external {
    type external;
    family inet {
      labeled-unicast {
        aigp;
      }
    }
  }
}

```

```

    }
  }
  export SET_EXPORT_ROUTES;
  peer-as 13979;
  neighbor 10.0.0.9;
}
}

user@PE7# show routing-options
router-id 10.9.9.7;
autonomous-system 7019;

```

### Verification

Confirm that the configuration is working properly.

- [Verifying That Device PE4 Is Receiving the AIGP Attribute from Its EBGp Neighbor PE2 on page 184](#)
- [Checking the IGP Metric on page 184](#)
- [Verifying That Device PE4 Adds the IGP Metric to the AIGP Attribute on page 185](#)
- [Verifying That Device PE7 Is Receiving the AIGP Attribute from Its EBGp Neighbor PE1 on page 185](#)
- [Verifying the Resolving AIGP Metric on page 186](#)
- [Verifying the Presence of AIGP Attributes in BGP Updates on page 189](#)

#### *Verifying That Device PE4 Is Receiving the AIGP Attribute from Its EBGp Neighbor PE2*

**Purpose** Make sure that the AIGP policy on Device PE2 is working.

**Action**

```

user@PE4> show route receive-protocol bgp 10.9.9.5 extensive
* 55.0.0.0/24 (1 entry, 1 announced)
  Accepted
  Route Label: 299888
  Nexthop: 10.100.1.5
  AS path: 7018 I
  AIGP: 20

* 99.0.0.0/24 (1 entry, 1 announced)
  Accepted
  Route Label: 299888
  Nexthop: 10.100.1.5
  AS path: 7018 I
  AIGP: 30

```

**Meaning** On Device PE2, the **aigp-originate** statement is configured with a distance of 20 (**aigp-originate distance 20**). This statement is applied to route 55.0.0.0/24. Likewise, the **aigp-originate distance 30** statement is applied to route 99.0.0.0/24. Thus, when Device PE4 receives these routes, the AIGP attribute is attached with the configured metrics.

#### *Checking the IGP Metric*

**Purpose** From Device PE4, check the IGP metric to the BGP next hop 10.100.1.5.

**Action** user@PE4> show route 10.100.1.5  
 inet.0: 30 destinations, 40 routes (30 active, 0 holddown, 0 hidden)  
 + = Active Route, - = Last Active, \* = Both

```

10.100.1.5/32      * [OSPF/10] 05:35:50, metric 2
                   > to 10.0.0.22 via fe-1/2/2.10
                   [BGP/170] 03:45:07, localpref 100, from 10.9.9.5
                   AS path: 7018 I
                   > to 10.0.0.22 via fe-1/2/2.10
  
```

**Meaning** The IGP metric for this route is 2.

#### *Verifying That Device PE4 Adds the IGP Metric to the AIGP Attribute*

**Purpose** Make sure that Device PE4 adds the IGP metric to the AIGP attribute when it readvertises routes to its IBGP neighbor, Device PE1.

**Action** user@PE4> show route advertising-protocol bgp 10.9.9.1 extensive

```

* 55.0.0.0/24 (1 entry, 1 announced)
  BGP group internal type Internal
    Route Label: 300544
    Nexthop: 10.100.1.4
    Flags: Nexthop Change
    Localpref: 100
    AS path: [13979] 7018 I
    AIGP: 22

* 99.0.0.0/24 (1 entry, 1 announced)
  BGP group internal type Internal
    Route Label: 300544
    Nexthop: 10.100.1.4
    Flags: Nexthop Change
    Localpref: 100
    AS path: [13979] 7018 I
    AIGP: 32
  
```

**Meaning** The IGP metric is added to the AIGP metric ( $20 + 2 = 22$  and  $30 + 2 = 32$ ), because the next hop is changed for these routes.

#### *Verifying That Device PE7 Is Receiving the AIGP Attribute from Its EBGp Neighbor PE1*

**Purpose** Make sure that the AIGP policy on Device PE1 is working.

**Action** user@PE7> show route receive-protocol bgp 10.0.0.9 extensive

\* 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

\* 99.0.0.0/24 (1 entry, 1 announced)

Accepted  
Route Label: 300112  
Nexthop: 10.0.0.9  
AS path: 13979 7018 I  
AIGP: 35

**Meaning** The 44.0.0.0/24 route is originated at Device PE4. The 55.0.0.0/24 and 99.0.0.0/24 routes are originated at Device PE2. The IGP distances are added to the configured AIGP distances.

#### *Verifying the Resolving AIGP Metric*

**Purpose** Confirm that if the prefix is resolved through recursion and the recursive next hops have AIGP metrics, the prefix has the sum of the AIGP values that are on the recursive BGP next hops.

**Action** 1. Add a static route to 66.0.0.0/24.

```
[edit routing-options]
user@PE2# set static route 66.0.0.0/24 discard
```

2. Delete the existing terms in the **aigp** policy statement on Device PE2.

```
[edit policy-options policy-statement aigp]
user@PE2# delete term 10
user@PE2# delete term 20
```

3. Configure a recursive route lookup for the route to 66.0.0.0.

The policy shows the AIGP metric for prefix 66.0.0.0/24 (none) and its recursive next hop. Prefix 66.0.0.0/24 is resolved by 55.0.0.1. Prefix 66.0.0.0/24 does not have its own AIGP metric being originated, but its recursive next hop, 55.0.0.1, has an AIGP value.

```
[edit policy-options policy-statement aigp]
user@PE2# set term 10 from route-filter 55.0.0.1/24 exact
user@PE2# set term 10 then aigp-originate distance 20
user@PE2# set term 10 then next-hop 10.100.1.5
user@PE2# set term 10 then accept
user@PE2# set term 20 from route-filter 66.0.0.0/24 exact
user@PE2# set term 20 then next-hop 55.0.0.1
```

**user@PE2# set term 20 then accept**

4. On Device PE4, run the **show route 55.0.0.0 extensive** command.

The value of Metric2 is the IGP metric to the BGP next hop. When Device PE4 readvertises these routes to its IBGP peer, Device PE1, the AIGP metric is the sum of AIGP + its Resolving AIGP metric + Metric2.

Prefix 55.0.0.0 shows its own IGP metric 20, as defined and advertised by Device PE2. It does not show a resolving AIGP value because it does not have a recursive BGP next hop. The value of Metric2 is 2.

```
user@PE4> show route 55.0.0.0 extensive
inet.0: 31 destinations, 41 routes (31 active, 0 holddown, 0 hidden)
55.0.0.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 55.0.0.0/24 -> {indirect(262151)}
Page 0 idx 0 Type 1 val 928d1b8
  Flags: Nexthop Change
  Nexthop: 10.100.1.4
  Localpref: 100
  AS path: [13979] 7018 I
  Communities:
  AIGP: 22
Path 55.0.0.0 from 10.9.9.5 Vector len 4. Val: 0
  *BGP   Preference: 170/-101
        Next hop type: Indirect
        Address: 0x925da38
        Next-hop reference count: 4
        Source: 10.9.9.5
        Next hop type: Router, Next hop index: 1004
        Next hop: 10.0.0.22 via fe-1/2/2.10, selected
        Label operation: Push 299888
        Label TTL action: prop-ttl
        Protocol next hop: 10.100.1.5
        Push 299888
        Indirect next hop: 93514d8 262151
        State: <Active Ext>
        Local AS: 13979 Peer AS: 7018
        Age: 22:03:26   Metric2: 2
        AIGP: 20
        Task: BGP_7018.10.9.9.5+58560
        Announcement bits (3): 3-KRT 4-BGP_RT_Background 5-Resolve tree 1
        AS path: 7018 I
        Accepted
        Route Label: 299888
        Localpref: 100
        Router ID: 10.9.9.5
        Indirect next hops: 1
          Protocol next hop: 10.100.1.5 Metric: 2
          Push 299888
          Indirect next hop: 93514d8 262151
          Indirect path forwarding next hops: 1
            Next hop type: Router
            Next hop: 10.0.0.22 via fe-1/2/2.10
            10.100.1.5/32 Originating RIB: inet.0
            Metric: 2                               Node path count: 1
            Forwarding nexthops: 1
              Nexthop: 10.0.0.22 via fe-1/2/2.10
```

5. On Device PE4, run the **show route 66.0.0.0 extensive** command.

Prefix 66.0.0.0/24 shows the Resolving AIGP, which is the sum of its own AIGP metric and its recursive BGP next hop:

66.0.0.1 = 0, 55.0.0.1 = 20, 0+20 = 20

```

user@PE4> show route 66.0.0.0 extensive
inet.0: 31 destinations, 41 routes (31 active, 0 holddown, 0 hidden)
66.0.0.0/24 (1 entry, 1 announced)
TSI:
KRT in-kerne1 66.0.0.0/24 -> {indirect(262162)}
Page 0 idx 0 Type 1 val 928cefc
  Flags: Nexthop Change
  Nexthop: 10.100.1.4
  Localpref: 100
  AS path: [13979] 7018 I
  Communities:
Path 66.0.0.0 from 10.9.9.5 Vector len 4. Val: 0
  *BGP Preference: 170/-101
    Next hop type: Indirect
    Address: 0x925d4e0
    Next-hop reference count: 4
    Source: 10.9.9.5
    Next hop type: Router, Next hop index: 1006
    Next hop: 10.0.0.22 via fe-1/2/2.10, selected
    Label operation: Push 299888, Push 299888(top)
    Label TTL action: prop-ttl, prop-ttl(top)
    Protocol next hop: 55.0.0.1
    Push 299888
    Indirect next hop: 9353e88 262162
    State: <Active Ext>
    Local AS: 13979 Peer AS: 7018
    Age: 31:42 Metric2:2
    Resolving-AIGP: 20
    Task: BGP_7018.10.9.9.5+58560
    Announcement bits (3): 3-KRT 4-BGP_RT_Background 5-Resolve tree 1
    AS path: 7018 I
    Accepted
    Route Label: 299888
    Localpref: 100
    Router ID: 10.9.9.5
    Indirect next hops: 1
      Protocol next hop: 55.0.0.1 Metric: 2 AIGP: 20
      Push 299888
      Indirect next hop: 9353e88 262162
      Indirect path forwarding next hops: 1
        Next hop type: Router
        Next hop: 10.0.0.22 via fe-1/2/2.10
      55.0.0.0/24 Originating RIB: inet.0
        Metric: 2 Node path count: 1
        Indirect nexthops: 1
          Protocol Nexthop: 10.100.1.5 Metric: 2 Push 299888
          Indirect nexthop: 93514d8 262151
          Indirect path forwarding nexthops: 1
            Nexthop: 10.0.0.22 via fe-1/2/2.10
          10.100.1.5/32 Originating RIB: inet.0
            Metric: 2 Node path count: 1
            Forwarding nexthops: 1
              Nexthop: 10.0.0.22 via fe-1/2/2.10

```

*Verifying the Presence of AIGP Attributes in BGP Updates*

**Purpose** If the AIGP attribute is not enabled under BGP (or the **group** or **neighbor** hierarchies), the AIGP attribute is silently discarded. Enable **traceoptions** and include the **packets** flag in the **detail** option in the configuration to confirm the presence of the AIGP attribute in transmitted or received BGP updates. This is useful when debugging AIGP issues.

**Action** 1. Configure Device PE2 and Device PE4 for **traceoptions**.

```
user@host> show protocols bgp
traceoptions {
  file bgp size 1m files 5;
  flag packets detail;
}
```

2. Check the **traceoptions** file on Device PE2.

The following sample shows Device PE2 advertising prefix 99.0.0.0/24 to Device PE4 (10.9.9.4) with an AIGP metric of 20:

```
user@PE2> show log bgp
Mar 22 09:27:18.982150 BGP SEND 10.9.9.5+49652 -> 10.9.9.4+179
Mar 22 09:27:18.982178 BGP SEND message type 2 (Update) length 70
Mar 22 09:27:18.982198 BGP SEND Update PDU length 70
Mar 22 09:27:18.982248 BGP SEND flags 0x40 code Origin(1): IGP
Mar 22 09:27:18.982273 BGP SEND flags 0x40 code ASPath(2) length 6: 7018
Mar 22 09:27:18.982295 BGP SEND flags 0x80 code AIGP(26): AIGP: 20
Mar 22 09:27:18.982316 BGP SEND flags 0x90 code MP_reach(14): AFI/SAFI 1/4
Mar 22 09:27:18.982341 BGP SEND      nhop 10.100.1.5 len 4
Mar 22 09:27:18.982372 BGP SEND    99.0.0.0/24 (label 301664)
Mar 22 09:27:33.665412 bgp_send: sending 19 bytes to abcd::10:255:170:84
(External AS 13979)
```

3. Verify that the route was received on Device PE4 using the **show route receive-protocol** command.

AIGP is not enabled on Device PE4, so the AIGP attribute is silently discarded for prefix 99.0.0.0/24 and does not appear in the following output:

```
user@PE4> show route receive-protocol bgp 10.9.9.5 extensive | find 55.0.0.0
* 99.0.0.0/24 (2 entries, 1 announced)
  Accepted
  Route Label: 301728
  Nexthop: 10.100.1.5
  AS path: 7018 I
```

4. Check the **traceoptions** file on Device PE4.

The following output from the **traceoptions** log shows that the 99.0.0.0/24 prefix was received with the AIGP attribute attached:

```
user@PE4> show log bgp
Mar 22 09:41:39.650295 BGP RECV 10.9.9.5+64690 -> 10.9.9.4+179
Mar 22 09:41:39.650331 BGP RECV message type 2 (Update) length 70
Mar 22 09:41:39.650350 BGP RECV Update PDU length 70
Mar 22 09:41:39.650370 BGP RECV flags 0x40 code Origin(1): IGP
Mar 22 09:41:39.650394 BGP RECV flags 0x40 code ASPath(2) length 6: 7018
Mar 22 09:41:39.650415 BGP RECV flags 0x80 code AIGP(26): AIGP: 20
Mar 22 09:41:39.650436 BGP RECV flags 0x90 code MP_reach(14): AFI/SAFI 1/4
Mar 22 09:41:39.650459 BGP RECV      nhop 10.100.1.5 len 4
```

```
Mar 22 09:41:39.650495 BGP RECV    99.0.0.0/24 (label 301728)
Mar 22 09:41:39.650574 bgp_rcv_nlri: 99.0.0.0/24
Mar 22 09:41:39.650607 bgp_rcv_nlri: 99.0.0.0/24 belongs to meshgroup
Mar 22 09:41:39.650629 bgp_rcv_nlri: 99.0.0.0/24 qualified bnp->ribact 0x0
12afcb 0x0
```

**Meaning** Performing this verification helps with AIGP troubleshooting and debugging issues. It enables you to verify which devices in your network send and receive AIGP attributes.

- Related Documentation**
- [Understanding BGP Path Selection on page 8](#)
  - [Examples: Configuring Internal BGP Peering on page 56](#)

---

## Example: Configuring AS Override

- [Understanding AS Override on page 190](#)
- [Example: Configuring a Layer 3 VPN with Route Reflection and AS Override on page 191](#)

### Understanding AS Override

The AS override feature allows a provider edge (PE) router to change the private autonomous system (AS) number used by a customer edge (CE) device on an external BGP (EBGP) session running on a VPN routing and forwarding (VRF) access link. The private AS number is changed to the PE AS number. Another CE device connected to another PE device sees the EBGP route coming from the first site with an AS path of provider-ASN provider-ASN, instead of provider-ASN site1-ASN. This allows enterprise networks to use the same private ASN on all sites.

The AS override feature offers a clear management advantage to the service provider because BGP by default does not accept BGP routes with an AS path attribute that contains the local AS number.

In an enterprise network with multiple sites, you might wish to use a single AS number across sites. Suppose, for example that two CE devices are in AS 64512 and that the provider network is in AS 65534.

When the service provider configures a Layer 3 VPN with this setup, even if the MPLS network has routes towards Device CE1 and Device CE2, Device CE1 and Device CE2 do not have routes to each other because the AS path attribute would appear as 64512 65534 64512. BGP uses the AS path attribute as its loop avoidance mechanism. If a site sees its own AS number more than once in the AS path, the route is considered invalid.

One way to overcome this difficulty is with the **as-override** statement, which is applied to the PE devices. The **as-override** statement replaces the CE device's AS number with that of the PE device, thus preventing the customer AS number from appearing more than once in the AS path attribute.

If a customer uses AS path prepending to make certain paths less desirable and the service provider uses AS override, each CE AS number occurrence in the AS-path is changed to the service provider AS number. For example, suppose that all customer sites

use the same AS number, say 64512. If the ISP uses AS number 65534, one customer site sees the path to another site as 65534 65534. If the customer prepends 64512 on a particular path to make it less desirable, another customer site sees that path as 65534 65534 64512 65534.

### Example: Configuring a Layer 3 VPN with Route Reflection and AS Override

Suppose that you are a service provider providing a managed MPLS-based Layer 3 VPN service. Your customer has several sites and requires BGP routing to customer edge (CE) devices at each site.

- [Requirements on page 191](#)
- [Overview on page 191](#)
- [Configuration on page 192](#)
- [Verification on page 199](#)

#### Requirements

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

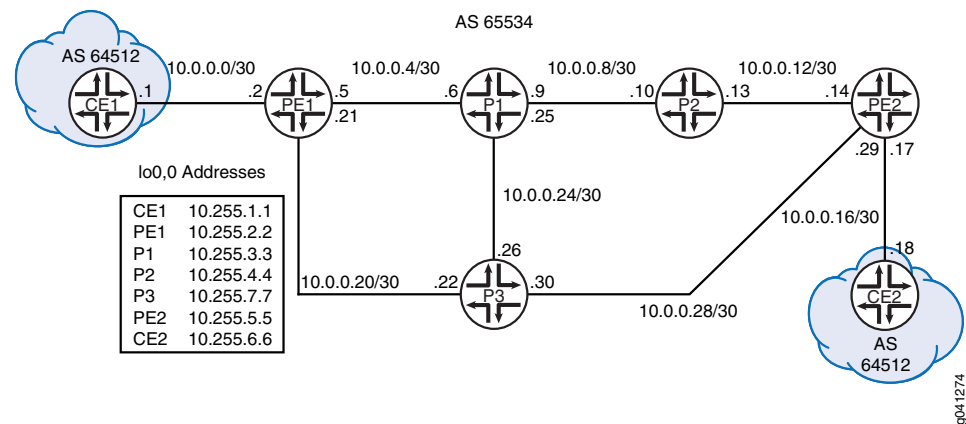
#### Overview

This example has two CE devices, two provider edge (PE) devices, and several provider core devices. The provider network is also using IS-IS to support LDP and BGP loopback reachability. Device P2 is acting as a route reflector (RR). Both CE devices are in autonomous system (AS) 64512. The provider network is in AS 65534.

The **as-override** statement is applied to the PE devices, thus replacing the CE device's AS number with that of the PE device. This prevents the customer AS number from appearing more than once in the AS path attribute.

Figure 24 on page 191 shows the topology used in this example.

Figure 24: AS Override Topology



[“CLI Quick Configuration” on page 192](#) shows the configuration for all of the devices in [Figure 24 on page 191](#). The section [“Step-by-Step Procedure” on page 195](#) describes the steps on Device PE1.

### Configuration

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CLI Quick Configuration</b> | 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 <b>[edit]</b> hierarchy level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Device CE1</b>              | <pre> set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.1/30 set interfaces ge-1/2/0 unit 0 family iso set interfaces lo0 unit 0 family inet address 10.255.1.1/32 set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0101.00 set protocols bgp group PE type external set protocols bgp group PE family inet unicast set protocols bgp group PE export ToBGP set protocols bgp group PE peer-as 65534 set protocols bgp group PE neighbor 10.0.0.2 set policy-options policy-statement ToBGP term Direct from protocol direct set policy-options policy-statement ToBGP term Direct then accept set routing-options router-id 10.255.1.1 set routing-options autonomous-system 64512 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Device P1</b>               | <pre> set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.6/30 set interfaces ge-1/2/0 unit 0 family iso set interfaces ge-1/2/0 unit 0 family mpls set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.9/30 set interfaces ge-1/2/1 unit 0 family iso set interfaces ge-1/2/1 unit 0 family mpls set interfaces ge-1/2/2 unit 0 family inet address 10.0.0.25/30 set interfaces ge-1/2/2 unit 0 family iso set interfaces ge-1/2/2 unit 0 family mpls set interfaces lo0 unit 0 family inet address 10.255.3.3/32 set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0303.00 set protocols mpls interface all set protocols mpls interface fxp0.0 disable set protocols bgp group l3vpn type internal set protocols bgp group l3vpn local-address 10.255.3.3 set protocols bgp group l3vpn family inet-vpn unicast set protocols bgp group l3vpn peer-as 65534 set protocols bgp group l3vpn local-as 65534 set protocols bgp group l3vpn neighbor 10.255.4.4 set protocols isis interface all level 2 metric 10 set protocols isis interface all level 1 disable set protocols isis interface fxp0.0 disable set protocols isis interface lo0.0 level 2 metric 0 set protocols ldp deaggregate set protocols ldp interface all set protocols ldp interface fxp0.0 disable set routing-options router-id 10.255.3.3 </pre> |
| <b>Device P2</b>               | <pre> set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.10/30 set interfaces ge-1/2/0 unit 0 family iso </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

```

set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.13/30
set interfaces ge-1/2/1 unit 0 family iso
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.4.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0404.00
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group Core-RRClients type internal
set protocols bgp group Core-RRClients local-address 10.255.4.4
set protocols bgp group Core-RRClients family inet-vpn unicast
set protocols bgp group Core-RRClients cluster 10.255.4.4
set protocols bgp group Core-RRClients peer-as 65534
set protocols bgp group Core-RRClients neighbor 10.255.3.3
set protocols bgp group Core-RRClients neighbor 10.255.7.7
set protocols bgp group Core-RRClients neighbor 10.255.2.2
set protocols bgp group Core-RRClients neighbor 10.255.5.5
set protocols isis interface all level 2 metric 10
set protocols isis interface all level 1 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set routing-options router-id 10.255.4.4
set routing-options autonomous-system 65534

```

**Device P3**

```

set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.22/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.26/30
set interfaces ge-1/2/1 unit 0 family iso
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces ge-1/2/2 unit 0 family inet address 10.0.0.30/30
set interfaces ge-1/2/2 unit 0 family iso
set interfaces ge-1/2/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.7.7/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0707.00
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group l3vpn type internal
set protocols bgp group l3vpn local-address 10.255.7.7
set protocols bgp group l3vpn family inet-vpn unicast
set protocols bgp group l3vpn peer-as 65534
set protocols bgp group l3vpn local-as 65534
set protocols bgp group l3vpn neighbor 10.255.4.4
set protocols isis interface all level 2 metric 10
set protocols isis interface all level 1 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set routing-options router-id 10.255.7.7

```

```
Device PE1  set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.2/30
             set interfaces ge-1/2/0 unit 0 family iso
             set interfaces ge-1/2/0 unit 0 family mpls
             set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.5/30
             set interfaces ge-1/2/1 unit 0 family iso
             set interfaces ge-1/2/1 unit 0 family mpls
             set interfaces ge-1/2/2 unit 0 family inet address 10.0.0.21/30
             set interfaces ge-1/2/2 unit 0 family iso
             set interfaces ge-1/2/2 unit 0 family mpls
             set interfaces lo0 unit 0 family inet address 10.255.2.2/32
             set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0202.00
             set protocols mpls interface ge-1/2/2.0
             set protocols mpls interface ge-1/2/1.0
             set protocols mpls interface lo0.0
             set protocols mpls interface fxp0.0 disable
             set protocols bgp group l3vpn type internal
             set protocols bgp group l3vpn local-address 10.255.2.2
             set protocols bgp group l3vpn family inet-vpn unicast
             set protocols bgp group l3vpn peer-as 65534
             set protocols bgp group l3vpn local-as 65534
             set protocols bgp group l3vpn neighbor 10.255.4.4
             set protocols isis interface ge-1/2/1.0 level 2 metric 10
             set protocols isis interface ge-1/2/1.0 level 1 disable
             set protocols isis interface ge-1/2/2.0 level 2 metric 10
             set protocols isis interface ge-1/2/2.0 level 1 disable
             set protocols isis interface fxp0.0 disable
             set protocols isis interface lo0.0 level 2 metric 0
             set protocols ldp deaggregate
             set protocols ldp interface ge-1/2/1.0
             set protocols ldp interface ge-1/2/2.0
             set protocols ldp interface fxp0.0 disable
             set protocols ldp interface lo0.0
             set routing-instances VPN-A instance-type vrf
             set routing-instances VPN-A interface ge-1/2/0.0
             set routing-instances VPN-A route-distinguisher 65534:1234
             set routing-instances VPN-A vrf-target target:65534:1234
             set routing-instances VPN-A protocols bgp group CE type external
             set routing-instances VPN-A protocols bgp group CE family inet unicast
             set routing-instances VPN-A protocols bgp group CE neighbor 10.0.0.1 peer-as 64512
             set routing-instances VPN-A protocols bgp group CE neighbor 10.0.0.1 as-override
             set routing-options router-id 10.255.2.2
             set routing-options autonomous-system 65534

Device PE2  set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.14/30
             set interfaces ge-1/2/0 unit 0 family iso
             set interfaces ge-1/2/0 unit 0 family mpls
             set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.17/30
             set interfaces ge-1/2/1 unit 0 family iso
             set interfaces ge-1/2/2 unit 0 family inet address 10.0.0.29/30
             set interfaces ge-1/2/2 unit 0 family iso
             set interfaces ge-1/2/2 unit 0 family mpls
             set interfaces lo0 unit 0 family inet address 10.255.5.5/32
             set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0505.00
             set protocols mpls interface ge-1/2/0.0
             set protocols mpls interface ge-1/2/2.0
```

```

set protocols mpls interface lo0.0
set protocols mpls interface fxp0.0 disable
set protocols bgp group l3vpn type internal
set protocols bgp group l3vpn local-address 10.255.5.5
set protocols bgp group l3vpn family inet-vpn unicast
set protocols bgp group l3vpn peer-as 65534
set protocols bgp group l3vpn local-as 65534
set protocols bgp group l3vpn neighbor 10.255.4.4
set protocols isis interface ge-1/2/0.0 level 2 metric 10
set protocols isis interface ge-1/2/0.0 level 1 disable
set protocols isis interface ge-1/2/2.0 level 2 metric 10
set protocols isis interface ge-1/2/2.0 level 1 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface ge-1/2/0.0
set protocols ldp interface ge-1/2/2.0
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface ge-1/2/1.0
set routing-instances VPN-A route-distinguisher 65534:1234
set routing-instances VPN-A vrf-target target:65534:1234
set routing-instances VPN-A protocols bgp group CE type external
set routing-instances VPN-A protocols bgp group CE family inet unicast
set routing-instances VPN-A protocols bgp group CE neighbor 10.0.0.18 peer-as 64512
set routing-instances VPN-A protocols bgp group CE neighbor 10.0.0.18 as-override
set routing-options router-id 10.255.5.5
set routing-options autonomous-system 65534

```

**Device CE2**

```

set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.18/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.255.6.6/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.0000.0606.00
set protocols bgp group PE type external
set protocols bgp group PE family inet unicast
set protocols bgp group PE export ToBGP
set protocols bgp group PE peer-as 65534
set protocols bgp group PE neighbor 10.0.0.17
set policy-options policy-statement ToBGP term Direct from protocol direct
set policy-options policy-statement ToBGP term Direct then accept
set routing-options router-id 10.255.6.6
set routing-options autonomous-system 64512

```

**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 AS override:

1. Configure the interfaces.

To enable MPLS, include the protocol family on the interface so that the interface does not discard incoming MPLS traffic.

[edit interfaces]

```

user@PE1# set ge-1/2/0 unit 0 family inet address 10.0.0.2/30
user@PE1# set ge-1/2/0 unit 0 family iso
user@PE1# set ge-1/2/0 unit 0 family mpls
user@PE1# set ge-1/2/1 unit 0 family inet address 10.0.0.5/30
user@PE1# set ge-1/2/1 unit 0 family iso
user@PE1# set ge-1/2/1 unit 0 family mpls
user@PE1# set ge-1/2/2 unit 0 family inet address 10.0.0.21/30
user@PE1# set ge-1/2/2 unit 0 family iso
user@PE1# set ge-1/2/2 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.2.2/32
user@PE1# set lo0 unit 0 family iso address 49.0001.0010.0000.0202.00

```

2. Add the interface to the MPLS protocol to establish the control plane level connectivity.

Set up the IGP so that the provider devices can communicate with each other.

To establish a mechanism to distribute MPLS labels, enable LDP. Optionally, for LDP, enable forwarding equivalence class (FEC) deaggregation, which results in faster global convergence.

```

[edit protocols]
user@PE1# set mpls interface ge-1/2/2.0
user@PE1# set mpls interface ge-1/2/1.0
user@PE1# set mpls interface lo0.0
user@PE1# set mpls interface fxp0.0 disable
user@PE1# set isis interface ge-1/2/1.0 level 2 metric 10
user@PE1# set isis interface ge-1/2/1.0 level 1 disable
user@PE1# set isis interface ge-1/2/2.0 level 2 metric 10
user@PE1# set isis interface ge-1/2/2.0 level 1 disable
user@PE1# set isis interface fxp0.0 disable
user@PE1# set isis interface lo0.0 level 2 metric 0
user@PE1# set ldp deaggregate
user@PE1# set ldp interface ge-1/2/1.0
user@PE1# set ldp interface ge-1/2/2.0
user@PE1# set ldp interface fxp0.0 disable
user@PE1# set ldp interface lo0.0

```

3. Enable the internal BGP (IBGP) connection to peer with the RR only, using the IPv4 VPN unicast address family.

```

[edit protocols bgp group l3vpn]
user@PE1# set type internal
user@PE1# set local-address 10.255.2.2
user@PE1# set family inet-vpn unicast
user@PE1# set peer-as 65534
user@PE1# set local-as 65534
user@PE1# set neighbor 10.255.4.4

```

4. Configure the routing instance, including the **as-override** statement.

Create the routing-instance (VRF) on the PE device, setting up the BGP configuration to peer with Device CE1.

```

[edit routing-instances VPN-A]
user@PE1# set instance-type vrf
user@PE1# set interface ge-1/2/0.0
user@PE1# set route-distinguisher 65534:1234

```

```

user@PE1# set vrf-target target:65534:1234
user@PE1# set protocols bgp group CE type external
user@PE1# set protocols bgp group CE family inet unicast
user@PE1# set protocols bgp group CE neighbor 10.0.0.1 peer-as 64512
user@PE1# set protocols bgp group CE neighbor 10.0.0.1 as-override

```

5. Configure the router ID and the AS number.

```

[edit routing-options]
user@PE1# set router-id 10.255.2.2
user@PE1# set autonomous-system 65534

```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **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@PE1# show interfaces
ge-1/2/0 {
  unit 2 {
    family inet {
      address 10.0.0.2/30;
    }
    family iso;
    family mpls;
  }
}
ge-1/2/1 {
  unit 5 {
    family inet {
      address 10.0.0.5/30;
    }
    family iso;
    family mpls;
  }
}
ge-1/2/2 {
  unit 21 {
    family inet {
      address 10.0.0.21/30;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.255.2.2/32;
    }
    family iso {
      address 49.0001.0010.0000.0202.00;
    }
  }
}

```

```
user@PE1# show protocols
mpls {
  interface ge-1/2/2.0;
  interface ge-1/2/1.0;
  interface lo0.0;
  interface fxp0.0 {
    disable;
  }
}
bgp {
  group l3vpn {
    type internal;
    local-address 10.255.2.2;
    family inet-vpn {
      unicast;
    }
    peer-as 65534;
    local-as 65534;
    neighbor 10.255.4.4;
  }
}
isis {
  interface ge-1/2/1.0 {
    level 2 metric 10;
    level 1 disable;
  }
  interface ge-1/2/2.0 {
    level 2 metric 10;
    level 1 disable;
  }
  interface fxp0.0 {
    disable;
  }
  interface lo0.0 {
    level 2 metric 0;
  }
}
ldp {
  deaggregate;
  interface ge-1/2/1.0;
  interface ge-1/2/2.0;
  interface fxp0.0 {
    disable;
  }
  interface lo0.0;
}

user@PE1# show routing-instances
VPN-A {
  instance-type vrf;
  interface ge-1/2/0.0;
  route-distinguisher 65534:1234;
  vrf-target target:65534:1234;
  protocols {
    bgp {
      group CE {
```

```

    type external;
    family inet {
        unicast;
    }
    neighbor 10.0.0.1 {
        peer-as 64512;
        as-override;
    }
}
}
}

user@PE1# show routing-options
router-id 10.255.2.2;
autonomous-system 65534;

```

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

### Verification

Confirm that the configuration is working properly.

- [Checking AS Path to the CE Devices on page 199](#)
- [Checking How the Route to Device CE2 Is Advertised on page 199](#)
- [Checking the Route on Device CE1 on page 200](#)

#### *Checking AS Path to the CE Devices*

**Purpose** Display information on Device PE1 about the AS path attribute for the route to Device CE2's loopback interface.

**Action** On Device PE1, from operational mode, enter the **show route table VPN-A.inet.0 10.255.6.6** command.

```
user@PE1> show route table VPN-A.inet.0 10.255.6.6
```

```
VPN-A.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.255.6.6/32      *[BGP/170] 02:19:35, localpref 100, from 10.255.4.4
                   AS path: 64512 I, validation-state: unverified
                   > to 10.0.0.22 via ge-1/2/2.0, Push 300032, Push 299776(top)

```

**Meaning** The output shows that Device PE1 has an AS path for 10.255.6.6/32 as coming from AS 64512.

#### *Checking How the Route to Device CE2 Is Advertised*

**Purpose** Make sure the route to Device CE2 is advertised to Device CE1 as if it is coming from the MPLS core.

**Action** On Device PE1, from operational mode, enter the **show route advertising-protocol bgp 10.0.0.1** command.

```
user@PE1> show route advertising-protocol bgp 10.0.0.1
```

```
VPN-A.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref    AS path
* 10.0.0.16/30          Self
* 10.255.1.1/32         10.0.0.1        65534 I
* 10.255.6.6/32         Self             65534 I
```

**Meaning** The output indicates that Device PE1 is advertising only its own AS number in the AS path.

#### *Checking the Route on Device CE1*

**Purpose** Make sure that Device CE1 contains only the provider AS number in the AS path for the route to Device CE2.

**Action** From operational mode, enter the **show route table inet.0 terse 10.255.6.6** command.

```
user@CE1> show route table inet.0 terse 10.255.6.6
```

```
inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

| A | V | Destination   | P | Prf | Metric 1 | Metric 2 | Next hop  | AS path     |
|---|---|---------------|---|-----|----------|----------|-----------|-------------|
| * | ? | 10.255.6.6/32 | B | 170 | 100      |          |           | 65534 65534 |
| I |   | unverified    |   |     |          |          | >10.0.0.2 |             |

**Meaning** The output indicates that Device CE1 has a route to Device CE2. The loop issue is resolved with the use of the **as-override** statement.

One route is hidden on the CE device. This is because Junos OS does not perform a BGP split horizon. Generally, split horizon in BGP is unnecessary, because any routes that might be received back by the originator are less preferred due to AS path length (for EBGp), AS path loop detection (IBGP), or other BGP metrics. Advertising routes back to the neighbor from which they were learned has a negligible effect on the router's performance, and is the correct thing to do.

**Related Documentation**

- [Examples: Configuring BGP Local AS on page 131](#)

## Example: Disabling Suppression of Route Advertisements

Junos OS does not advertise the routes learned from one EBGp peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGp peers that are in the same autonomous system (AS) as the originating peer, regardless of the routing instance. You can modify this behavior by including the **advertise-peer-as** statement in the configuration.

If you include the **advertise-peer-as** statement in the configuration, BGP advertises the route regardless of this check.

To restore the default behavior, include the **no-advertise-peer-as** statement in the configuration:

```
no-advertise-peer-as;
```

The route suppression default behavior is disabled if the **as-override** statement is included in the configuration. If you include both the **as-override** and **no-advertise-peer-as** statements in the configuration, the **no-advertise-peer-as** statement is ignored.

- [Requirements on page 201](#)
- [Overview on page 201](#)
- [Configuration on page 202](#)
- [Verification on page 206](#)

## Requirements

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

## Overview

This example shows three routing devices with external BGP (EBGP) connections. Device R2 has an EBGP connection to Device R1 and another EBGP connection to Device R3. Although separated by Device R2 which is in AS 64511, Device R1 and Device R3 are in the same AS (AS 64512). Device R1 and Device R3 advertise into BGP direct routes to their own loopback interface addresses.

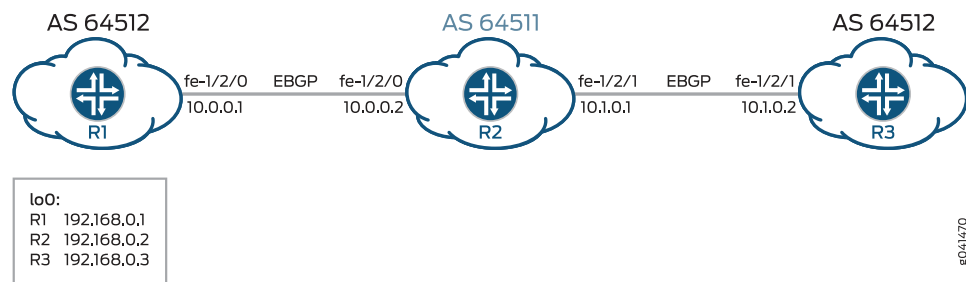
Device R2 receives these loopback interface routes, and the **advertise peer-as** statement allows Device R2 to advertise them. Specifically, Device R1 sends the 192.168.0.1 route to Device R2, and because Device R2 has the **advertise peer-as** configured, Device R2 can send the 192.168.0.1 route to Device R3. Likewise, Device R3 sends the 192.168.0.3 route to Device R2, and **advertise peer-as** enables Device R2 to forward the route to Device R1.

To enable Device R1 and Device R3 to accept routes that contain their own AS number in the AS path, the **loops 2** statement is required on Device R1 and Device R3.

## Topology

[Figure 25 on page 202](#) shows the sample network.

Figure 25: BGP Topology for advertise-peer-as



“CLI Quick Configuration” on page 202 shows the configuration for all of the devices in Figure 25 on page 202.

The section “Step-by-Step Procedure” on page 203 describes the steps on Device R1 and 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 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp family inet unicast loops 2
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 300

Device R2
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group ext type external
set protocols bgp group ext advertise-peer-as
set protocols bgp group ext export send-direct
set protocols bgp group ext neighbor 10.0.0.1 peer-as 300
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 200

Device R3
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp family inet unicast loops 2
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.1.0.1

```

```

set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 300

```

### 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 Device R1:

1. Configure the device interfaces.

```

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

user@R1# set lo0 unit 0 family inet address 192.168.0.1/32

```

2. Configure BGP.

```

[edit protocols bgp group ext]
user@R1# set type external
user@R1# set peer-as 200
user@R1# set neighbor 10.0.0.2

```

3. Prevent routes from Device R3 from being hidden on Device R1 by including the **loops 2** statement.

The **loops 2** statement means that the local device's own AS number can appear in the AS path up to one time without causing the route to be hidden. The route is hidden if the local device's AS number is detected in the path two or more times.

```

[edit protocols bgp family inet unicast]
user@R1# set loops 2

```

4. Configure the routing policy that sends direct routes.

```

[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept

```

5. Apply the export policy to the BGP peering session with Device R2.

```

[edit protocols bgp group ext]
user@R1# set export send-direct

```

6. Configure the autonomous system (AS) number.

```

[edit routing-options ]
user@R1# set autonomous-system 300

```

### 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 Device R2:

1. Configure the device interfaces.

```

[edit interfaces]

```

```
user@R2# set fe-1/2/0 unit 0 family inet address 10.0.0.2/30
```

```
user@R2# set fe-1/2/1 unit 0 family inet address 10.1.0.1/30
```

```
user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure BGP.

```
[edit protocols bgp group ext]
```

```
user@R2# set type external
```

```
user@R2# set neighbor 10.0.0.1 peer-as 300
```

```
user@R2# set neighbor 10.1.0.2 peer-as 300
```

3. Configure Device R2 to advertise routes learned from one EBGP peer to another EBGP peer in the same AS.

In other words, advertise to Device R1 routes learned from Device R3 (and the reverse), even though Device R1 and Device R3 are in the same AS.

```
[edit protocols bgp group ext]
```

```
user@R2# set advertise-peer-as
```

4. Configure a routing policy that sends direct routes.

```
[edit policy-options policy-statement send-direct term 1]
```

```
user@R2# set from protocol direct
```

```
user@R2# set then accept
```

5. Apply the export policy.

```
[edit protocols bgp group ext]
```

```
user@R2# set export send-direct
```

6. Configure the AS number.

```
[edit routing-options]
```

```
user@R2# set autonomous-system 200
```

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

```
Device R1 user@R1# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
```

```

user@R1# show protocols
bgp {
  family inet {
    unicast {
      loops 2;
    }
  }
  group ext {
    type external;
    export send-direct;
    peer-as 200;
    neighbor 10.0.0.2;
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show routing-options
autonomous-system 300;

```

**Device R2**

```

user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    family inet {
      address 10.1.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.2/32;
    }
  }
}

user@R2# show protocols
bgp {
  group ext {
    type external;
    advertise-peer-as;
    export send-direct;
    neighbor 10.0.0.1 {
      peer-as 300;
    }
  }
}

```

```

    }
    neighbor 10.1.0.2 {
        peer-as 300;
    }
}

user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R2# show routing-options
autonomous-system 200;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the BGP Routes

**Purpose** Make sure that the routing tables on Device R1 and Device R3 contain the expected routes.

**Action** 1. On Device R2, deactivate the **advertise-peer-as** statement in the BGP configuration.

```

[edit protocols bgp group ext]
user@R2# deactivate advertise-peer-as
user@R2# commit

```

2. On Device R3, deactivate the **loops** statement in the BGP configuration.

```

[edit protocols bgp family inet unicast ]
user@R3# deactivate unicast loops
user@R3# commit

```

3. On Device R1, check to see what routes are advertised to Device R2.

```

user@R1> show route advertising-protocol bgp 10.0.0.2
inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref    AS path
* 10.0.0.0/30       Self              0
* 192.168.0.1/32    Self              0

```

4. On Device R2, check to see what routes are received from Device R1.

```

user@R2> show route receive-protocol bgp 10.0.0.1
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref    AS path
10.0.0.0/30         10.0.0.1         0
* 192.168.0.1/32    10.0.0.1         0

```

5. On Device R2, check to see what routes are advertised to Device R3.

```

user@R2> show route advertising-protocol bgp 10.1.0.2
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref    AS path

```

```
* 10.0.0.0/30          Self          I
* 10.1.0.0/30          Self          I
* 192.168.0.2/32       Self          I
```

6. On Device R2, activate the **advertise-peer-as** statement in the BGP configuration.

```
[edit protocols bgp group ext]
user@R2# activate advertise-peer-as
user@R2# commit
```

7. On Device R2, recheck the routes that are advertised to Device R3.

```
user@R2> show route advertising-protocol bgp 10.1.0.2
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref  AS path
* 10.0.0.0/30       Self              I
* 10.1.0.0/30       Self              I
* 192.168.0.1/32    Self              300 I
* 192.168.0.2/32    Self              I
* 192.168.0.3/32    10.1.0.2         300 I
```

8. On Device R3, check the routes that are received from Device R2.

```
user@R3> show route receive-protocol bgp 10.1.0.1
inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref  AS path
* 10.0.0.0/30       10.1.0.1         200 I
  10.1.0.0/30       10.1.0.1         200 I
* 192.168.0.2/32    10.1.0.1         200 I
```

9. On Device R3, activate the **loops** statement in the BGP configuration.

```
[edit protocols bgp family inet unicast ]
user@R3# activate unicast loops
user@R3# commit
```

10. On Device R3, recheck the routes that are received from Device R2.

```
user@R3> show route receive-protocol bgp 10.1.0.1
inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 1 hidden)
  Prefix            Nexthop          MED      Lclpref  AS path
* 10.0.0.0/30       10.1.0.1         200 I
  10.1.0.0/30       10.1.0.1         200 I
* 192.168.0.1/32    10.1.0.1         200 300 I
* 192.168.0.2/32    10.1.0.1         200 I
```

**Meaning** First the **advertise-peer-as** statement and the **loops** statement are deactivated so that the default behavior can be examined. Device R1 sends to Device R2 a route to Device R1's loopback interface address, 192.168.0.1/32. Device R2 does not advertise this route to Device R3. After activating the **advertise-peer-as** statement, Device R2 does advertise the 192.168.0.1/32 route to Device R3. Device R3 does not accept this route until after the **loops** statement is activated.

**Related Documentation**

- [Example: Configuring a Layer 3 VPN with Route Reflection and AS Override on page 191](#)

## Configuring 4-Byte Autonomous System Numbers

---

This section describes how to configure a 4-byte AS number and how to verify if the BGP peer supports 4-byte AS numbers.

The AS number can be specified in plain number format or in AS-dot notation format on routers running Junos OS Release 9.2 and later. For example, the 4-byte AS number of 65,546 is represented in plain-number format as 65546. The same AS number is represented in AS-dot notation format as 1.10 on routers running Junos OS Release 9.2 and later.

- To configure a 4-byte AS number in AS-dot notation format, include the **autonomous-system** statement and specify the 4-byte AS number. In the following example the AS number is set to 1.10.

```
user@host# set routing-options autonomous-system 1.10
```

- To configure a 4-byte AS number in plain number format, include the **autonomous-system** statement and specify the 4-byte AS number. In the following example the AS number is set to 65546.

```
user@host# set routing-options autonomous-system 65546
```

- After a BGP peer session has been negotiated, you can verify whether the peer supports 4-byte AS numbers or not. To verify whether the peer supports 4-byte AS numbers or not, use the **show bgp neighbor** command. In the following example the peer does not support 4-byte AS numbers.

```
user@host# show bgp neighbor 192.168.1.9 | match "AS"
Peer: 192.168.1.9+179 AS 65056 Local: 192.168.1.3+52616 AS 65000
Peer does not support 4 byte AS extension
```

- In the following example the peer does support 4-byte AS numbers.

```
user@host# show bgp neighbor 192.168.1.9 | match "AS"
Peer: 192.168.1.10+52679 AS 1000000000 Local: 192.168.1.3+179 AS 65000
Peer supports 4 byte AS extension (peer-as 1000000000)
```

### Related Documentation

- [4-Byte Autonomous System Numbers Overview on page 17](#)
- [Configuring 4-Byte AS Numbers and BGP Extended Community Attributes on page 21](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 2-Byte AS Number on page 27](#)
- [Establishing a Peer Relationship Between a 4-Byte Capable Router and a 2-Byte Capable Router Using a 4-Byte AS Number on page 28](#)
- [Juniper Networks Implementation of 4-Byte Autonomous System Numbers on page 18](#)
- [Prepending 4-Byte AS Numbers in an AS Path on page 20](#)
- [Understanding 4-Byte AS Numbers and Route Distinguishers on page 24](#)
- [Understanding 4-Byte AS Numbers and Route Loop Detection on page 25](#)

- [Understanding a 4-Byte Capable Router AS Path Through a 2-Byte Capable Domain on page 22](#)
- [Disabling Attribute Set Messages on Independent AS Domains for BGP Loop Detection on page 209](#)

## Disabling Attribute Set Messages on Independent AS Domains for BGP Loop Detection

BGP loop detection for a specific route uses the local autonomous system (AS) domain for the routing instance. By default, all routing instances belong to a single primary routing instance domain. Therefore, BGP loop detection uses the local ASs configured on all of the routing instances. Depending on your network configuration, this default behavior can cause routes to be looped and hidden.

To limit the local ASs in the primary routing instance, you can configure an independent AS domain for a routing instance. The independent domain is separate from the primary routing instance and keeps the AS paths of the independent domain from being shared with the AS path and the AS path attributes of other domains.

By default, independent domains use transitive path attribute 128 (attribute set) messages to tunnel the independent domain's BGP attributes through the internal BGP (IBGP) core. However, the attribute set message behavior for independent domains is undesired in many cases. If you only want to configure independent domains to maintain the independence of local ASs in the routing instance, and perform BGP loop detection only for the specified local ASs in the routing instance, you can disable the attribute set messages.

To disable attribute set messages on an independent domain, include the **independent-domain no-attrset** statement:

1. Select the routing instance that contains the independent domain you want to modify. You can select the routing instance from the following hierarchy levels:
  - **[edit routing-instances *routing-instance-name*]**
  - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**
2. Disable attribute set messages on the independent domain.

```
[edit routing-instances instance-name]
user@host# set routing-options autonomous-system as-number independent-domain
no-attrset
```



**TIP:** When you disable attribute set messages, we recommend that you specify the AS number of the primary routing instance. This ensures that the primary routing instance AS is treated as a local AS in the routing instance and is used for BGP loop detection.

After you specify a routing instance for an independent domain, the local ASs are only associated with that routing instance. That means BGP loop detection uses only the local ASs defined in the routing instance.

**Related  
Documentation**

- *autonomous-system*
- *independent-domain*
- [Example: Configuring a Local AS for EBGp Sessions on page 136](#)

## CHAPTER 6

# BGP Policy Configuration

- [Example: Applying Routing Policies at Different Levels of the BGP Hierarchy on page 211](#)
- [Example: Configuring BGP Interactions with IGPs on page 220](#)
- [Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS on page 223](#)
- [Example: Configuring BGP Route Advertisement on page 233](#)
- [Example: Configuring EBGp Multihop on page 241](#)
- [Example: Configuring BGP Route Preference \(Administrative Distance\) on page 250](#)
- [Example: Configuring BGP Path Selection on page 257](#)
- [Example: Removing Private AS Numbers on page 267](#)
- [Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers on page 274](#)
- [Example: Configuring Conditional Installation of Prefixes in a Routing Table on page 278](#)
- [Example: Setting BGP to Advertise Inactive Routes on page 298](#)
- [Example: Configuring BGP to Advertise the Best External Route to Internal Peers on page 304](#)
- [Example: Disabling Suppression of Route Advertisements on page 312](#)
- [Example: Defining a Routing Policy That Removes BGP Communities on page 319](#)
- [Example: Defining a Routing Policy Based on the Number of BGP Communities on page 326](#)
- [Example: Using Routing Policy to Set a Preference Value for BGP Routes on page 333](#)
- [Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths on page 339](#)

### Example: Applying Routing Policies at Different Levels of the BGP Hierarchy

This example shows BGP configured in a simple network topology and explains how routing policies take effect when they are applied at different levels of the BGP configuration.

- [Requirements on page 212](#)
- [Overview on page 212](#)

- [Configuration on page 213](#)
- [Verification on page 217](#)

## Requirements

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

## Overview

For BGP, you can apply policies as follows:

- BGP global **import** and **export** statements—Include these statements at the **[edit protocols bgp]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances routing-instance-name protocols bgp]** hierarchy level).
- Group **import** and **export** statements—Include these statements at the **[edit protocols bgp group group-name]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances routing-instance-name protocols bgp group group-name]** hierarchy level).
- Peer **import** and **export** statements—Include these statements at the **[edit protocols bgp group group-name neighbor address]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances routing-instance-name protocols bgp group group-name neighbor address]** hierarchy level).

A peer-level **import** or **export** statement overrides a group **import** or **export** statement. A group-level **import** or **export** statement overrides a global BGP **import** or **export** statement.

In this example, a policy named **send-direct** is applied at the global level, another policy named **send-192.168.0.1** is applied at the group level, and a third policy named **send-192.168.20.1** is applied at the neighbor level.

```
user@host# show protocols
bgp {
  local-address 1.1.1.1;
  export send-direct;
  group internal-peers {
    type internal;
    export send-192.168.0.1;
    neighbor 2.2.2.2 {
      export send-192.168.20.1;
    }
    neighbor 3.3.3.3;
  }
  group other-group {
    type internal;
    neighbor 4.4.4.4;
  }
}
```

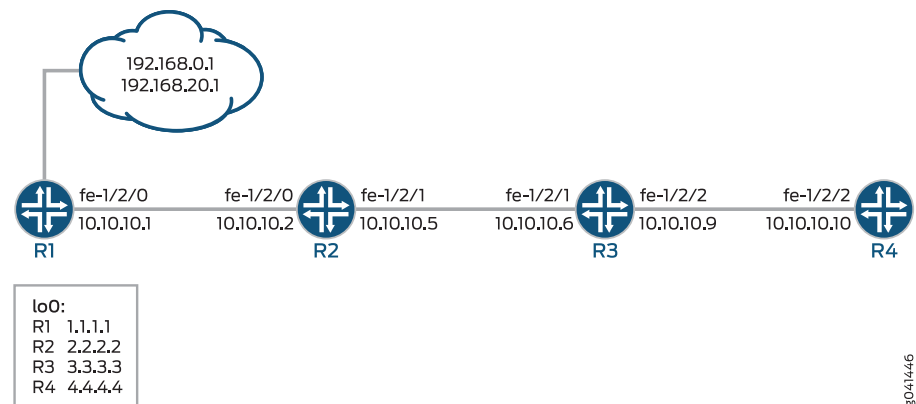
A key point, and one that is often misunderstood and that can lead to problems, is that in such a configuration, only the most explicit policy is applied. A neighbor-level policy is more explicit than a group-level policy, which in turn is more explicit than a global policy.

The neighbor 2.2.2.2 is subjected only to the send-192.168.20.1 policy. The neighbor 3.3.3.3, lacking anything more specific, is subjected only to the send-192.168.0.1 policy. Meanwhile, neighbor 4.4.4.4 in group other-group has no group or neighbor-level policy, so it uses the send-direct policy.

If you need to have neighbor 2.2.2.2 perform the function of all three policies, you can write and apply a new neighbor-level policy that encompasses the functions of the other three, or you can apply all three existing policies, as a chain, to neighbor 2.2.2.2.

Figure 26 on page 213 shows the sample network.

**Figure 26: Applying Routing Policies to BGP**



“CLI Quick Configuration” on page 213 shows the configuration for all of the devices in Figure 26 on page 213.

The section “Step-by-Step Procedure” on page 215 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 0 description to-R2
set interfaces fe-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set protocols bgp local-address 1.1.1.1
set protocols bgp export send-direct
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers export send-static-192.168.0
set protocols bgp group internal-peers neighbor 2.2.2.2 export send-static-192.168.20
set protocols bgp group internal-peers neighbor 3.3.3.3
set protocols bgp group other-group type internal
set protocols bgp group other-group neighbor 4.4.4.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static-192.168.0 term 1 from protocol static
```

```
set policy-options policy-statement send-static-192.168.0 term 1 from route-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement send-static-192.168.0 term 1 then accept
set policy-options policy-statement send-static-192.168.20 term 1 from protocol static
set policy-options policy-statement send-static-192.168.20 term 1 from route-filter
  192.168.20.0/24 orlonger
set policy-options policy-statement send-static-192.168.20 term 1 then accept
set routing-options static route 192.168.0.1/32 discard
set routing-options static route 192.168.20.1/32 discard
set routing-options router-id 1.1.1.1
set routing-options autonomous-system 17
```

Device R2

```
set interfaces fe-1/2/0 unit 0 description to-R1
set interfaces fe-1/2/0 unit 0 family inet address 10.10.10.2/30
set interfaces fe-1/2/1 unit 0 description to-R3
set interfaces fe-1/2/1 unit 0 family inet address 10.10.10.5/30
set interfaces lo0 unit 0 family inet address 2.2.2.2/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 2.2.2.2
set protocols bgp group internal-peers neighbor 3.3.3.3
set protocols bgp group internal-peers neighbor 1.1.1.1
set protocols bgp group internal-peers neighbor 4.4.4.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set routing-options router-id 2.2.2.2
set routing-options autonomous-system 17
```

Device R3

```
set interfaces fe-1/2/1 unit 0 description to-R2
set interfaces fe-1/2/1 unit 0 family inet address 10.10.10.6/30
set interfaces fe-1/2/2 unit 0 description to-R4
set interfaces fe-1/2/2 unit 0 family inet address 10.10.10.9/30
set interfaces lo0 unit 0 family inet address 3.3.3.3/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 3.3.3.3
set protocols bgp group internal-peers neighbor 2.2.2.2
set protocols bgp group internal-peers neighbor 1.1.1.1
set protocols bgp group internal-peers neighbor 4.4.4.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set routing-options router-id 3.3.3.3
set routing-options autonomous-system 17
```

Device R4

```
set interfaces fe-1/2/2 unit 0 description to-R3
set interfaces fe-1/2/2 unit 0 family inet address 10.10.10.10/30
set interfaces lo0 unit 0 family inet address 4.4.4.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 4.4.4.4
set protocols bgp group internal-peers neighbor 2.2.2.2
set protocols bgp group internal-peers neighbor 1.1.1.1
set protocols bgp group internal-peers neighbor 3.3.3.3
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set routing-options router-id 4.4.4.4
```

**set routing-options autonomous-system 17**

**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 an IS-IS default route policy:

1. Configure the device interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 0 description to-R2
user@R1# set fe-1/2/0 unit 0 family inet address 10.10.10.1/30
```

```
user@R1# set lo0 unit 0 family inet address 1.1.1.1/32
```

2. Enable OSPF, or another interior gateway protocols (IGP), on the interfaces.

```
[edit protocols OSPF area 0.0.0.0]
user@R1# set interface lo0.0 passive
user@R1# set interface fe-1/2/0.0
```

3. Configure static routes.

```
[edit routing-options]
user@R1# set static route 192.168.0.1/32 discard
user@R1# set static route 192.168.20.1/32 discard
```

4. Enable the routing policies.

```
[edit protocols policy-options]
user@R1# set policy-statement send-direct term 1 from protocol direct
user@R1# set policy-statement send-direct term 1 then accept
```

```
user@R1# set policy-statement send-static-192.168.0 term 1 from protocol static
user@R1# set policy-statement send-static-192.168.0 term 1 from route-filter
192.168.0.0/24 orlonger
user@R1# set policy-statement send-static-192.168.0 term 1 then accept
```

```
user@R1# set policy-statement send-static-192.168.20 term 1 from protocol static
user@R1# set policy-statement send-static-192.168.20 term 1 from route-filter
192.168.20.0/24 orlonger
user@R1# set policy-statement send-static-192.168.20 term 1 then accept
```

5. Configure BGP and apply the export policies.

```
[edit protocols bgp]
user@R1# set local-address 1.1.1.1
user@R1# set group internal-peers type internal
user@R1# set group internal-peers export send-static-192.168.0
user@R1# set group internal-peers neighbor 2.2.2.2 export send-static-192.168.20
user@R1# set group internal-peers neighbor 3.3.3.3
user@R1# set group other-group type internal
user@R1# set group other-group neighbor 4.4.4.4
```

6. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
```

```
user@R1# set router-id 1.1.1.1
user@R1# set autonomous-system 17
```

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

```
[edit]
user@R1# commit
```

---

## Results

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 0 {
    description to-R2;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}

user@R1# show protocols
bgp {
  local-address 1.1.1.1;
  export send-direct;
  group internal-peers {
    type internal;
    export send-static-192.168.0;
    neighbor 2.2.2.2 {
      export send-static-192.168.20;
    }
    neighbor 3.3.3.3;
  }
  group other-group {
    type internal;
    neighbor 4.4.4.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/2/0.0;
```

```

    }
  }
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement send-static-192.168.0 {
  term 1 {
    from {
      protocol static;
      route-filter 192.168.0.0/24 orlonger;
    }
    then accept;
  }
}
policy-statement send-static-192.168.20 {
  term 1 {
    from {
      protocol static;
      route-filter 192.168.20.0/24 orlonger;
    }
    then accept;
  }
}

user@R1# show routing-options
static {
  route 192.168.0.1/32 discard;
  route 192.168.20.1/32 discard;
}
router-id 1.1.1.1;
autonomous-system 17;

```

## Verification

Confirm that the configuration is working properly.

- [Verifying BGP Route Learning on page 217](#)
- [Verifying BGP Route Receiving on page 219](#)

### Verifying BGP Route Learning

**Purpose** Make sure that the BGP export policies are working as expected by checking the routing tables.

**Action** user@R1> **show route protocol direct**

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

```
1.1.1.1/32      *[Direct/0] 1d 22:19:47
                 > via lo0.0
10.10.10.0/30   *[Direct/0] 1d 22:19:47
                 > via fe-1/2/0.0
```

user@R1> **show route protocol static**

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

```
192.168.0.1/32  *[Static/5] 02:20:03
                 Discard
192.168.20.1/32 *[Static/5] 02:20:03
                 Discard
```

user@R2> **show route protocol bgp**

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

```
192.168.20.1/32  *[BGP/170] 02:02:40, localpref 100, from 1.1.1.1
                 AS path: I, validation-state: unverified
                 > to 10.10.10.1 via fe-1/2/0.0
```

user@R3> **show route protocol bgp**

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

```
192.168.0.1/32  *[BGP/170] 02:02:51, localpref 100, from 1.1.1.1
                 AS path: I, validation-state: unverified
                 > to 10.10.10.5 via fe-1/2/1.0
```

user@R4> **show route protocol bgp**

```
inet.0: 9 destinations, 11 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
1.1.1.1/32      [BGP/170] 1d 20:38:54, localpref 100, from 1.1.1.1
                 AS path: I, validation-state: unverified
                 > to 10.10.10.9 via fe-1/2/2.0
10.10.10.0/30   [BGP/170] 1d 20:38:54, localpref 100, from 1.1.1.1
                 AS path: I, validation-state: unverified
                 > to 10.10.10.9 via fe-1/2/2.0
```

**Meaning** On Device R1, the **show route protocol direct** command displays two direct routes: 1.1.1.1/32 and 10.10.10.0/30. The **show route protocol static** command displays two static routes: 192.168.0.1/32 and 192.168.20.1/32.

On Device R2, the **show route protocol bgp** command shows that the only route that Device R2 has learned through BGP is the 192.168.20.1/32 route.

On Device R3, the **show route protocol bgp** command shows that the only route that Device R3 has learned through BGP is the 192.168.0.1/32 route.

On Device R4, the **show route protocol bgp** command shows that the only routes that Device R4 has learned through BGP are the 1.1.1.1/32 and 10.10.10.0/30 routes.

### Verifying BGP Route Receiving

**Purpose** Make sure that the BGP export policies are working as expected by checking the BGP routes received from Device R1.

**Action** user@R2> **show route receive-protocol bgp 1.1.1.1**

```
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref    AS path
* 192.168.20.1/32   1.1.1.1          100      100        I
```

user@R3> **show route receive-protocol bgp 1.1.1.1**

```
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref    AS path
* 192.168.0.1/32    1.1.1.1          100      100        I
```

user@R4> **show route receive-protocol bgp 1.1.1.1**

```
inet.0: 9 destinations, 11 routes (9 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref    AS path
1.1.1.1/32         1.1.1.1          100      100        I
10.10.10.0/30      1.1.1.1          100      100        I
```

**Meaning** On Device R2, the **route receive-protocol bgp 1.1.1.1** command shows that Device R2 received only one BGP route, 192.168.20.1/32, from Device R1.

On Device R3, the **route receive-protocol bgp 1.1.1.1** command shows that Device R3 received only one BGP route, 192.168.0.1/32, from Device R1.

On Device R4, the **route receive-protocol bgp 1.1.1.1** command shows that Device R4 received two BGP routes, 1.1.1.1/32 and 10.10.10.0/30, from Device R1.

In summary, when multiple policies are applied at different CLI hierarchies in BGP, only the most specific application is evaluated, to the exclusion of other, less specific policy applications. Although this point might seem to make sense, it is easily forgotten during router configuration, when you mistakenly believe that a neighbor-level policy is combined with a global or group-level policy, only to find that your policy behavior is not as anticipated.

- Related Documentation**
- *Example: Configuring Policy Chains and Route Filters*
  - *Example: Configuring a Policy Subroutine*
  - *Example: Configuring Routing Policy Prefix Lists*
  - [export on page 646](#)
  - [import on page 667](#)

## Example: Configuring BGP Interactions with IGPs

---

- [Understanding Routing Policies on page 220](#)
- [Example: Injecting OSPF Routes into the BGP Routing Table on page 220](#)

### Understanding Routing Policies

Each routing policy is identified by a policy name. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks. Each routing policy name must be unique within a configuration.

Once a policy is created and named, it must be applied before it is active. You apply routing policies using the **import** and **export** statements at the **protocols>protocol-name** level in the configuration hierarchy.

In the **import** statement, you list the name of the routing policy to be evaluated when routes are imported into the routing table from the routing protocol.

In the **export** statement, you list the name of the routing policy to be evaluated when routes are being exported from the routing table into a dynamic routing protocol. Only active routes are exported from the routing table.

To specify more than one policy and create a policy chain, you list the policies using a space as a separator. If multiple policies are specified, the policies are evaluated in the order in which they are specified. As soon as an accept or reject action is executed, the policy chain evaluation ends.

### Example: Injecting OSPF Routes into the BGP Routing Table

This example shows how to create a policy that injects OSPF routes into the BGP routing table.

- [Requirements on page 220](#)
- [Overview on page 221](#)
- [Configuration on page 221](#)
- [Verification on page 223](#)
- [Troubleshooting on page 223](#)

---

#### Requirements

Before you begin:

- Configure network interfaces.
- Configure external peer sessions. See [“Example: Configuring External BGP Point-to-Point Peer Sessions” on page 34](#).
- Configure interior gateway protocol (IGP) sessions between peers.

## Overview

In this example, you create a routing policy called **injectpolicy1** and a routing term called **injectterm1**. The policy injects OSPF routes into the BGP routing table.

## Configuration

- [Configuring the Routing Policy on page 221](#)
- [Configuring Tracing for the Routing Policy on page 222](#)

### Configuring the Routing Policy

**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.

```
set policy-options policy-statement injectpolicy1 term injectterm1 from protocol ospf
set policy-options policy-statement injectpolicy1 term injectterm1 from area 0.0.0.1
set policy-options policy-statement injectpolicy1 term injectterm1 then accept
set protocols bgp export injectpolicy1
```

**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 inject OSPF routes into a BGP routing table:

1. Create the policy term.  

```
[edit policy-options policy-statement injectpolicy1]
user@host# set term injectterm1
```
2. Specify OSPF as a match condition.  

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set from protocol ospf
```
3. Specify the routes from an OSPF area as a match condition.  

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set from area 0.0.0.1
```
4. Specify that the route is to be accepted if the previous conditions are matched.  

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set then accept
```
5. Apply the routing policy to BGP.  

```
[edit]
user@host# set protocols bgp export injectpolicy1
```

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

```
user@host# show policy-options
policy-statement injectpolicy1 {
  term injectterm1 {
    from {
      protocol ospf;
      area 0.0.0.1;
    }
    then accept;
  }
}
```

```
user@host# show protocols bgp
export injectpolicy1;
```

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

### *Configuring Tracing for the Routing Policy*

**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.

```
set policy-options policy-statement injectpolicy1 term injectterm1 then trace
set routing-options traceoptions file ospf-bgp-policy-log
set routing-options traceoptions file size 5m
set routing-options traceoptions file files 5
set routing-options traceoptions flag policy
```

**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*.

1. Include a trace action in the policy.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# then trace
```

2. Configure the tracing file for the output.

```
[edit routing-options traceoptions]
user@host# set file ospf-bgp-policy-log
user@host# set file size 5m
user@host# set file files 5
user@host# set flag policy
```

**Results** Confirm your configuration by entering the **show policy-options** and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement injectpolicy1 {
  term injectterm1 {
    then {
      trace;
    }
  }
}
```

```

    }
  }

user@host# show routing-options
traceoptions {
  file ospf-bgp-policy-log size 5m files 5;
  flag policy;
}

```

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

### Verification

Confirm that the configuration is working properly.

#### *Verifying That the Expected BGP Routes Are Present*

**Purpose** Verify the effect of the export policy.

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

### Troubleshooting

- [Using the show log Command to Examine the Actions of the Routing Policy on page 223](#)

#### *Using the show log Command to Examine the Actions of the Routing Policy*

**Problem** The routing table contains unexpected routes, or routes are missing from the routing table.

**Solution** If you configure policy tracing as shown in this example, you can run the **show log ospf-bgp-policy-log** command to diagnose problems with the routing policy. The **show log ospf-bgp-policy-log** command displays information about the routes that the **injectpolicy1** policy term analyzes and acts upon.

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS

- [Understanding BGP Communities and Extended Communities as Routing Policy Match Conditions on page 223](#)
- [Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS on page 225](#)

## Understanding BGP Communities and Extended Communities as Routing Policy Match Conditions

A *BGP community* is a group of destinations that share a common property. Community information is included as a path attribute in BGP update messages. This information identifies community members and enables you to perform actions on a group without

having to elaborate upon each member. You can use community and extended communities attributes to trigger routing decisions, such as acceptance, rejection, preference, or redistribution.

You can assign community tags to non-BGP routes through configuration (for static, aggregate, or generated routes) or an import routing policy. These tags can then be matched when BGP exports the routes.

A community value is a 32-bit field that is divided into two main sections. The first 16 bits of the value encode the AS number of the network that originated the community, while the last 16 bits carry a unique number assigned by the AS. This system attempts to guarantee a globally unique set of community values for each AS in the Internet. Junos OS uses a notation of **as-number:community-value**, where each value is a decimal number. The AS values of 0 and 65,535 are reserved, as are all of the community values within those AS numbers. Each community, or set of communities, is given a name within the **[edit policy-options]** configuration hierarchy. The name of the community uniquely identifies it to the routing device and serves as the method by which routes are categorized. For example, a route with a community value of 64510:1111 might belong to the community named **AS64510-routes**. The community name is also used within a routing policy as a match criterion or as an action. The command syntax for creating a community is: **policy-options community name members [community-ids]**. The **community-ids** are either a single community value or multiple community values. When more than one value is assigned to a community name, the routing device interprets this as a logical AND of the community values. In other words, a route must have all of the configured values before being assigned the community name.

The regular community attribute is four octets. Networking enhancements, such as VPNs, have functionality requirements that can be satisfied by an attribute such as a community. However, the 4-octet community value does not provide enough expansion and flexibility to accommodate VPN requirements. This leads to the creation of extended communities. An extended community is an 8-octet value that is also divided into two main sections. The first 2 octets of the community encode a type field while the last 6 octets carry a unique set of data in a format defined by the type field. Extended communities provide a larger range for grouping or categorizing communities.

The BGP extended communities attribute format has three fields:

**type:administrator:assigned-number**. The routing device expects you to use the words **target** or **origin** to represent the type field. The administrator field uses a decimal number for the AS or an IPv4 address, while the assigned number field expects a decimal number no larger than the size of the field (65,535 for 2 octets or 4,294,967,295 for 4 octets).

When specifying community IDs for standard and extended community attributes, you can use UNIX-style regular expressions. The only exception is for VPN import policies (**vrf-import**), which do not support regular expressions for the extended communities attribute.

## Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS

This example defines a policy that takes BGP routes from the **Edu** community and places them into IS-IS with a metric of 63.

- [Requirements on page 225](#)
- [Overview on page 225](#)
- [Configuration on page 226](#)
- [Verification on page 232](#)

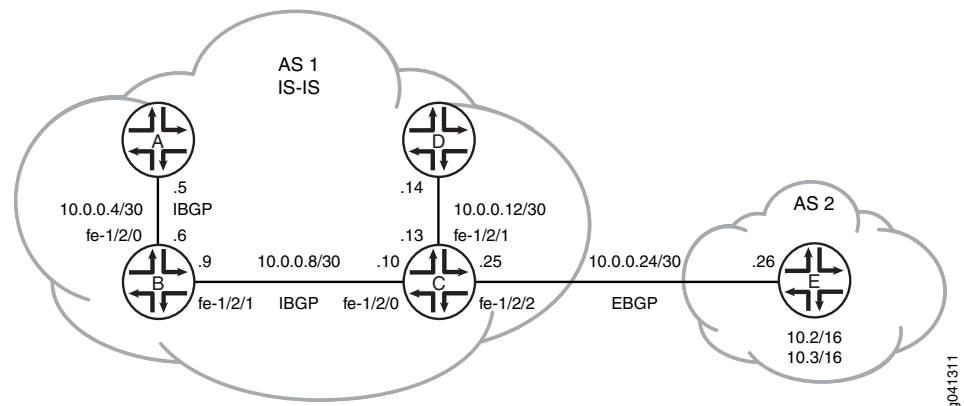
### Requirements

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

### Overview

[Figure 27 on page 225](#) shows the topology used in this example.

**Figure 27: Redistributing BGP Routes with a Specific Community Tag into IS-IS**



In this example, Device A, Device B, Device C, and Device D are in autonomous system (AS) 1 and are running IS-IS. All of the AS 1 devices, except Device D, are running internal BGP (IBGP).

Device E is in AS 2 and has an external BGP (EBGP) peering session with Device C. Device E has two static routes, 10.2.0.0/16 and 10.3.0.0/16. These routes are tagged with the Edu 2:5 community attribute and are advertised by way of EBGP to Device C.

Device C accepts the BGP routes that are tagged with the Edu 2:5 community attribute, redistributes the routes into IS-IS, and applies an IS-IS metric of 63 to these routes.

[“CLI Quick Configuration” on page 226](#) shows the configuration for all of the devices in [Figure 27 on page 225](#). The section [“Step-by-Step Procedure” on page 227](#) describes the steps on Device C and Device E.

## Configuration

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CLI Quick Configuration</b> | 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 <b>[edit]</b> hierarchy level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Device A</b>                | <pre> set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.5/30 set interfaces fe-1/2/0 unit 0 family iso set interfaces lo0 unit 0 family inet address 192.168.0.1/32 set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0001.00 set protocols bgp group int type internal set protocols bgp group int local-address 192.168.0.1 set protocols bgp group int neighbor 192.168.0.2 set protocols bgp group int neighbor 192.168.0.3 set protocols isis interface fe-1/2/0.0 level 1 disable set protocols isis interface lo0.0 set routing-options router-id 192.168.0.1 set routing-options autonomous-system 1 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Device B</b>                | <pre> set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.6/30 set interfaces fe-1/2/0 unit 0 family iso set interfaces fe-1/2/1 unit 0 family inet address 10.0.0.9/30 set interfaces fe-1/2/1 unit 0 family iso set interfaces lo0 unit 0 family inet address 192.168.0.2/32 set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0002.00 set protocols bgp group int type internal set protocols bgp group int local-address 192.168.0.2 set protocols bgp group int neighbor 192.168.0.1 set protocols bgp group int neighbor 192.168.0.3 set protocols isis interface fe-1/2/0.0 level 1 disable set protocols isis interface fe-1/2/1.0 level 1 disable set protocols isis interface lo0.0 set routing-options router-id 192.168.0.2 set routing-options autonomous-system 1 </pre>                                                                                                                                                                                                                                                                 |
| <b>Device C</b>                | <pre> set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.10/30 set interfaces fe-1/2/0 unit 0 family iso set interfaces fe-1/2/1 unit 0 family inet address 10.0.0.13/30 set interfaces fe-1/2/1 unit 0 family iso set interfaces fe-1/2/2 unit 0 family inet address 10.0.0.25/30 set interfaces fe-1/2/2 unit 0 family iso set interfaces lo0 unit 0 family inet address 192.168.0.3/32 set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0003.00 set protocols bgp group int type internal set protocols bgp group int local-address 192.168.0.3 set protocols bgp group int neighbor 192.168.0.1 set protocols bgp group int neighbor 192.168.0.2 set protocols bgp group external-peers type external set protocols bgp group external-peers export send-isis-and-direct set protocols bgp group external-peers peer-as 2 set protocols bgp group external-peers neighbor 10.0.0.26 set protocols isis export Edu-to-isis set protocols isis interface fe-1/2/0.0 level 1 disable set protocols isis interface fe-1/2/1.0 level 1 disable </pre> |

```

set protocols isis interface fe-1/2/2.0 level 1 disable
set protocols isis interface fe-1/2/2.0 level 2 passive
set protocols isis interface lo0.0
set policy-options policy-statement Edu-to-isis term 1 from protocol bgp
set policy-options policy-statement Edu-to-isis term 1 from community Edu
set policy-options policy-statement Edu-to-isis term 1 then metric 63
set policy-options policy-statement Edu-to-isis term 1 then accept
set policy-options policy-statement send-isis-and-direct term 1 from protocol isis
set policy-options policy-statement send-isis-and-direct term 1 from protocol direct
set policy-options policy-statement send-isis-and-direct term 1 from route-filter
  10.0.0.0/16 orlonger
set policy-options policy-statement send-isis-and-direct term 1 from route-filter
  192.168.0.0/16 orlonger
set policy-options policy-statement send-isis-and-direct term 1 then accept
set policy-options community Edu members 2:5
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 1

```

**Device D**

```

set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.14/30
set interfaces fe-1/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.0.4/32
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0004.00
set protocols isis interface fe-1/2/0.0 level 1 disable
set protocols isis interface lo0.0
set routing-options router-id 192.168.0.4
set routing-options autonomous-system 1

```

**Device E**

```

set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.26/30
set interfaces lo0 unit 7 family inet address 192.168.0.5/32 primary
set interfaces lo0 unit 7 family inet address 10.2.0.1/32
set interfaces lo0 unit 7 family inet address 10.3.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers export statics
set protocols bgp group external-peers peer-as 1
set protocols bgp group external-peers neighbor 10.0.0.25
set policy-options policy-statement statics from protocol static
set policy-options policy-statement statics then community add Edu
set policy-options policy-statement statics then accept
set policy-options community Edu members 2:5
set routing-options static route 10.2.0.0/16 reject
set routing-options static route 10.2.0.0/16 install
set routing-options static route 10.3.0.0/16 reject
set routing-options static route 10.3.0.0/16 install
set routing-options router-id 192.168.0.5
set routing-options autonomous-system 2

```

**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 Device E:

1. Configure the interfaces.  
[edit interfaces]

```

user@E# set fe-1/2/0 unit 0 family inet address 10.0.0.26/30
user@E# set lo0 unit 7 family inet address 192.168.0.5/32 primary
user@E# set lo0 unit 7 family inet address 10.2.0.1/32
user@E# set lo0 unit 7 family inet address 10.3.0.1/32

```

2. Configure the **statics** policy, which adds the **Edu** community attribute to the static routes.

```

[edit policy-options]
user@E# set policy-statement statics from protocol static
user@E# set policy-statement statics then community add Edu
user@E# set policy-statement statics then accept
user@E# set community Edu members 2:5

```

3. Configure EBGp and apply the **statics** policy.

```

[edit protocols bgp group external-peers]
user@E# set type external
user@E# set export statics
user@E# set peer-as 1
user@E# set protocols bgp group external-peers neighbor 10.0.0.25

```

4. Configure the static routes.

```

[edit routing-options static]
user@E# set route 10.2.0.0/16 reject
user@E# set route 10.2.0.0/16 install
user@E# set route 10.3.0.0/16 reject
user@E# set route 10.3.0.0/16 install

```

5. Configure the router ID and the AS number.

```

[edit routing-options]
user@E# set router-id 192.168.0.5
user@E# set autonomous-system 2

```

#### 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 Device C:

1. Configure the interfaces.

```

[edit interfaces]
user@C# set fe-1/2/0 unit 0 family inet address 10.0.0.10/30
user@C# set fe-1/2/0 unit 0 family iso
user@C# set fe-1/2/1 unit 0 family inet address 10.0.0.13/30
user@C# set fe-1/2/1 unit 0 family iso
user@C# set fe-1/2/2 unit 0 family inet address 10.0.0.25/30
user@C# set fe-1/2/2 unit 0 family iso
user@C# set lo0 unit 0 family inet address 192.168.0.3/32
user@C# set lo0 unit 0 family iso address 49.0002.0192.0168.0003.00

```

2. Configure IBGP.

```

[edit protocols bgp group int]
user@C# set type internal
user@C# set local-address 192.168.0.3

```

```

user@C# set neighbor 192.168.0.1
user@C# set neighbor 192.168.0.2

```

3. Configure the Edu-to-isis policy, which redistributes the Edu-tagged BGP routes learned from Device E and applies a metric of 63.

```

[edit policy-options]
user@C# set policy-statement Edu-to-isis term 1 from protocol bgp
user@C# set policy-statement Edu-to-isis term 1 from community Edu
user@C# set policy-statement Edu-to-isis term 1 then metric 63
user@C# set policy-statement Edu-to-isis term 1 then accept
user@C# set community Edu members 2:5

```

4. Enable IS-IS on the interfaces, and apply the Edu-to-isis policy.

```

[edit protocols isis]
user@C# set export Edu-to-isis
user@C# set interface fe-1/2/0.0 level 1 disable
user@C# set interface fe-1/2/1.0 level 1 disable
user@C# set interface fe-1/2/2.0 level 1 disable
user@C# set interface fe-1/2/2.0 level 2 passive
user@C# set interface lo0.0

```

5. Configure the send-isis-and-direct policy, which redistributes routes to Device E, through EBGP.

Without this policy, Device E would not have connectivity to the networks in AS 1.

```

[edit policy-options policy-statement send-isis-and-direct term 1]
user@C# set from protocol isis
user@C# set from protocol direct
user@C# set from route-filter 10.0.0.0/16 orlonger
user@C# set from route-filter 192.168.0.0/16 orlonger
user@C# set then accept

```

6. Configure EBGP and apply the send-isis-and-direct policy.

```

[edit protocols bgp group external-peers]
user@C# set type external
user@C# set export send-isis-and-direct
user@C# set peer-as 2
user@C# set neighbor 10.0.0.26

```

7. Configure the router ID and the autonomous system (AS) number.

```

[edit routing-options]
user@C# set router-id 192.168.0.3
user@C# set autonomous-system 1

```

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

**Device E**

```

user@E# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.26/30;
    }
  }
}

```

```
    }  
  }  
}  
lo0 {  
  unit 0 {  
    family inet {  
      address 192.168.0.5/32 {  
        primary;  
      }  
      address 10.2.0.1/32;  
      address 10.3.0.1/32;  
    }  
  }  
}
```

user@E# show protocols

```
bgp {  
  group external-peers {  
    type external;  
    export statics;  
    peer-as 1;  
    neighbor 10.0.0.25;  
  }  
}
```

user@E# show policy-options

```
policy-statement statics {  
  from protocol static;  
  then {  
    community add Edu;  
    accept;  
  }  
}
```

community Edu members 2:5;

user@E# show routing-options

```
static {  
  route 10.2.0.0/16 {  
    reject;  
    install;  
  }  
  route 10.3.0.0/16 {  
    reject;  
    install;  
  }  
}
```

router-id 192.168.0.5;

autonomous-system 2;

**Device C** user@C# show interfaces

```
fe-1/2/0 {  
  unit 0 {  
    family inet {  
      address 10.0.0.10/30;  
    }  
    family iso;  
  }  
}
```

```

}
fe-1/2/1 {
  unit 0 {
    family inet {
      address 10.0.0.13/30;
    }
    family iso;
  }
}
fe-1/2/2 {
  unit 0 {
    family inet {
      address 10.0.0.25/30;
    }
    family iso;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.3/32;
    }
    family iso {
      address 49.0002.0192.0168.0003.00;
    }
  }
}
}

```

user@C# show protocols

```

bgp {
  group int {
    type internal;
    local-address 192.168.0.3;
    neighbor 192.168.0.1;
    neighbor 192.168.0.2;
  }
  group external-peers {
    type external;
    export send-isis-and-direct;
    peer-as 2;
    neighbor 10.0.0.26;
  }
}
isis {
  export Edu-to-isis;
  interface fe-1/2/0.0 {
    level 1 disable;
  }
  interface fe-1/2/1.0 {
    level 1 disable;
  }
  interface fe-1/2/2.0 {
    level 1 disable;
    level 2 passive;
  }
  interface lo0.0;
}

```

```

}
user@C# show policy-options
policy-statement Edu-to-isis {
  term 1 {
    from {
      protocol bgp;
      community Edu;
    }
    then {
      metric 63;
      accept;
    }
  }
}
policy-statement send-isis-and-direct {
  term 1 {
    from {
      protocol [ isis direct ];
      route-filter 10.0.0.0/16 orlonger;
      route-filter 192.168.0.0/16 orlonger;
    }
    then accept;
  }
}
community Edu members 2:5;

user@C# show routing-options
router-id 192.168.0.3;
autonomous-system 1;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the IS-IS Neighbor

**Purpose** Verify that the BGP routes from Device E are communicated on the IS-IS network in AS 1.

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

```

user@D> show route protocol isis
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[IS-IS/18] 22:30:53, metric 30
                 > to 10.0.0.13 via fe-1/2/0.0
10.0.0.8/30      *[IS-IS/18] 22:30:53, metric 20
                 > to 10.0.0.13 via fe-1/2/0.0
10.0.0.24/30     *[IS-IS/18] 03:31:21, metric 20
                 > to 10.0.0.13 via fe-1/2/0.0
10.2.0.0/16      *[IS-IS/165] 02:36:31, metric 73
                 > to 10.0.0.13 via fe-1/2/0.0
10.3.0.0/16      *[IS-IS/165] 02:36:31, metric 73
                 > to 10.0.0.13 via fe-1/2/0.0

```

```

192.168.0.1/32    *[IS-IS/18] 03:40:28, metric 30
                  > to 10.0.0.13 via fe-1/2/0.0
192.168.0.2/32    *[IS-IS/18] 22:30:53, metric 20
                  > to 10.0.0.13 via fe-1/2/0.0
192.168.0.3/32    *[IS-IS/18] 22:30:53, metric 10
                  > to 10.0.0.13 via fe-1/2/0.0

```

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

**Meaning** As expected, the 10.2.0.0/16 and 10.3.0.0/16 routes are in Device D's routing table as IS-IS external routes with a metric of 73. If Device C had not added 63 to the metric, Device D would have a metric of 10 for these routes.

**Related Documentation**

- [Example: Redistributing OSPF Routes into IS-IS](#)

## Example: Configuring BGP Route Advertisement

- [Understanding Route Advertisement on page 233](#)
- [Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 237](#)

## Understanding Route Advertisement

All routing protocols use the Junos OS routing table to store the routes that they learn and to determine which routes they should advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table. For information about routing policy, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.

When configuring BGP routing policy, you can perform the following tasks:

- [Applying Routing Policy on page 233](#)
- [Setting BGP to Advertise Inactive Routes on page 234](#)
- [Configuring BGP to Advertise the Best External Route to Internal Peers on page 235](#)
- [Configuring How Often BGP Exchanges Routes with the Routing Table on page 236](#)
- [Disabling Suppression of Route Advertisements on page 237](#)

### Applying Routing Policy

You define routing policy at the **[edit policy-options]** hierarchy level. To apply policies you have defined for BGP, include the **import** and **export** statements within the BGP configuration.

You can apply policies as follows:

- BGP global **import** and **export** statements—Include these statements at the **[edit protocols bgp]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances routing-instance-name protocols bgp]** hierarchy level).
- Group **import** and **export** statements—Include these statements at the **[edit protocols bgp group group-name]** hierarchy level (for routing instances, include these statements

at the [edit routing-instances *routing-instance-name* protocols bgp group *group-name*] hierarchy level).

- Peer **import** and **export** statements—Include these statements at the [edit protocols bgp group *group-name* neighbor *address*] hierarchy level (for routing instances, include these statements at the [edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*] hierarchy level).

A peer-level **import** or **export** statement overrides a group **import** or **export** statement. A group-level **import** or **export** statement overrides a global BGP **import** or **export** statement.

To apply policies, see the following sections:

- [Applying Policies to Routes Being Imported into the Routing Table from BGP on page 234](#)
- [Applying Policies to Routes Being Exported from the Routing Table into BGP on page 234](#)

#### ***Applying Policies to Routes Being Imported into the Routing Table from BGP***

To apply policy to routes being imported into the routing table from BGP, include the **import** statement, listing the names of one or more policies to be evaluated:

```
import [ policy-names ];
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching filter is applied to the route. If no match is found, BGP places into the routing table only those routes that were learned from BGP routing devices.

#### ***Applying Policies to Routes Being Exported from the Routing Table into BGP***

To apply policy to routes being exported from the routing table into BGP, include the **export** statement, listing the names of one or more policies to be evaluated:

```
export [ policy-names ];
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching filter is applied to the route. If no routes match the filters, the routing table exports into BGP only the routes that it learned from BGP.

---

#### ***Setting BGP to Advertise Inactive Routes***

By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. To have the routing table export to BGP the best route learned by BGP even if Junos OS did not select it to be an active route, include the **advertise-inactive** statement:

```
advertise-inactive;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

### Configuring BGP to Advertise the Best External Route to Internal Peers

In general, deployed BGP implementations do not advertise the external route with the highest local preference value to internal peers unless it is the best route. Although this behavior was required by an earlier version of the BGP version 4 specification, RFC 1771, it was typically not followed in order to minimize the amount of advertised information and to prevent routing loops. However, there are scenarios in which advertising the best external route is beneficial, in particular, situations that can result in IBGP route oscillation.

In Junos OS Release 9.3 and later, you can configure BGP to advertise the best external route into an internal BGP (IBGP) mesh group, a route reflector cluster, or an autonomous system (AS) confederation, even when the best route is an internal route.



**NOTE:** In order to configure the `advertise-external` statement on a route reflector, you must disable intracluster reflection with the `no-client-reflect` statement.

When a routing device is configured as a route reflector for a cluster, a route advertised by the route reflector is considered internal if it is received from an internal peer with the same cluster identifier or if both peers have no cluster identifier configured. A route received from an internal peer that belongs to another cluster, that is, with a different cluster identifier, is considered external.

In a confederation, when advertising a route to a confederation border router, any route from a different confederation sub-AS is considered external.

You can also configure BGP to advertise the external route only if the route selection process reaches the point where the multiple exit discriminator (MED) metric is evaluated. As a result, an external route with an AS path worse (that is, longer) than that of the active path is not advertised.

Junos OS also provides support for configuring a BGP export policy that matches on the state of an advertised route. You can match on either active or inactive routes. For more information, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*.

To configure BGP to advertise the best external path to internal peers, include the **`advertise-external`** statement:

```
advertise-external;
```



**NOTE:** The `advertise-external` statement is supported at both the group and neighbor level. If you configure the statement at the neighbor level, you must configure it for all neighbors in a group. Otherwise, the group is automatically split into different groups.

For a complete list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

To configure BGP to advertise the best external path only if the route selection process reaches the point where the MED value is evaluated, include the **conditional** statement:

```
advertise-external {  
  conditional;  
}
```

### Configuring How Often BGP Exchanges Routes with the Routing Table

---

BGP stores the route information it receives from update messages in the routing table, and the routing table exports active routes from the routing table into BGP. BGP then advertises the exported routes to its peers. By default, the exchange of route information between BGP and the routing table occurs immediately after the routes are received. This immediate exchange of route information might cause instabilities in the network reachability information. To guard against this, you can delay the time between when BGP and the routing table exchange route information.

To configure how often BGP and the routing table exchange route information, include the **out-delay** statement:

```
out-delay seconds;
```

By default, the routing table retains some of the route information learned from BGP. To have the routing table retain all or none of this information, include the **keep** statement:

```
keep (all | none);
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

The routing table can retain the route information learned from BGP in one of the following ways:

- Default (omit the **keep** statement)—Keep all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.
- **keep all**—Keep all route information that was learned from BGP.
- **keep none**—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking, such as AS path or next hop. When you configure **keep none** for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.

In an AS path healing situation, routes with looped paths theoretically could become usable during a soft reconfiguration when the AS path loop limit is changed. However, there is a significant memory usage difference between the default and **keep all**.

Consider the following scenarios:

- A peer readvertises routes back to the peer from which it learned them.

This can happen in the following cases:

- Another vendor's routing device advertises the routes back to the sending peer.

- The Junos OS peer's default behavior of not readvertising routes back to the sending peer is overridden by configuring **advertise-peer-as**.
- A provider edge (PE) routing device discards any VPN route that does not have any of the expected route targets.

When **keep all** is configured, the behavior of discarding routes received in the above scenarios is overridden.

### Disabling Suppression of Route Advertisements

Junos OS does not advertise the routes learned from one EBGp peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGp peers that are in the same AS as the originating peer, regardless of the routing instance. You can modify this behavior by including the **advertise-peer-as** statement in the configuration. To disable the default advertisement suppression, include the **advertise-peer-as** statement:

```
advertise-peer-as;
```



**NOTE:** The route suppression default behavior is disabled if the **as-override** statement is included in the configuration.

If you include the **advertise-peer-as** statement in the configuration, BGP advertises the route regardless of this check.

To restore the default behavior, include the **no-advertise-peer-as** statement in the configuration:

```
no-advertise-peer-as;
```

If you include both the **as-override** and **no-advertise-peer-as** statements in the configuration, the **no-advertise-peer-as** statement is ignored. You can include these statements at multiple hierarchy levels.

For a list of hierarchy levels at which you can include these statements, see the statement summary section for these statements.

### Example: Configuring BGP Prefix-Based Outbound Route Filtering

This example shows how to configure a Juniper Networks router to accept route filters from remote peers and perform outbound route filtering using the received filters.

- [Requirements on page 238](#)
- [Overview on page 238](#)
- [Configuration on page 238](#)
- [Verification on page 240](#)

## Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol (IGP).

## Overview

You can configure a BGP peer to accept route filters from remote peers and perform outbound route filtering using the received filters. By filtering out unwanted updates, the sending peer saves resources needed to generate and transmit updates, and the receiving peer saves resources needed to process updates. This feature can be useful, for example, in a virtual private network (VPN) in which subsets of customer edge (CE) devices are not capable of processing all the routes in the VPN. The CE devices can use prefix-based outbound route filtering to communicate to the provider edge (PE) routing device to transmit only a subset of routes, such as routes to the main data centers only.

The maximum number of prefix-based outbound route filters that a BGP peer can accept is 5000. If a remote peer sends more than 5000 outbound route filters to a peer address, the additional filters are discarded, and a system log message is generated.

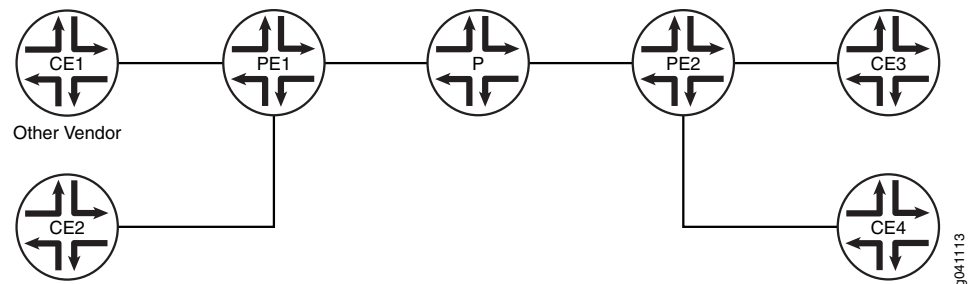
You can configure interoperability for the routing device as a whole or for specific BGP groups or peers only.

## Topology

In the sample network, Device CE1 is a router from another vendor. The configuration shown in this example is on Juniper Networks Router PE1.

Figure 28 on page 238 shows the sample network.

Figure 28: BGP Prefix-Based Outbound Route Filtering



## 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.

```

PE1    set protocols bgp group cisco-peers type external
       set protocols bgp group cisco-peers description "to CE1"
  
```

```

set protocols bgp group cisco-peers local-address 192.168.165.58
set protocols bgp group cisco-peers peer-as 35
set protocols bgp group cisco-peers outbound-route-filter bgp-orf-cisco-mode
set protocols bgp group cisco-peers outbound-route-filter prefix-based accept inet
set protocols bgp group cisco-peers neighbor 192.168.165.56
set routing-options autonomous-system 65500

```

**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 Router PE1 to accept route filters from Device CE1 and perform outbound route filtering using the received filters:

1. Configure the local autonomous system.

```

[edit routing-options]
user@PE1# set autonomous-system 65500

```

2. Configure external peering with Device CE1.

```

[edit protocols bgp group cisco-peers]
user@PE1# set type external
user@PE1# set description "to CE1"
user@PE1# set local-address 192.168.165.58
user@PE1# set peer-as 35
user@PE1# set neighbor 192.168.165.56

```

3. Configure Router PE1 to accept IPv4 route filters from Device CE1 and perform outbound route filtering using the received filters.

```

[edit protocols bgp group cisco-peers]
user@PE1# set outbound-route-filter prefix-based accept inet

```

4. (Optional) Enable interoperability with routing devices that use the vendor-specific compatibility code of 130 for outbound route filters and the code type of 128.

The IANA standard code is 3, and the standard code type is 64.

```

[edit protocols bgp group cisco-peers]
user@PE1# set outbound-route-filter bgp-orf-cisco-mode

```

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

```

user@PE1# show protocols
group cisco-peers {
  type external;
  description "to CE1";
  local-address 192.168.165.58;
  peer-as 35;
  outbound-route-filter {
    bgp-orf-cisco-mode;
    prefix-based {
      accept {
        inet;

```

```

    }
  }
}
neighbor 192.168.165.56;
}

user@PE1# show routing-options
autonomous-system 65500;

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying the Outbound Route Filter on page 240](#)
- [Verifying the BGP Neighbor Mode on page 240](#)

#### *Verifying the Outbound Route Filter*

**Purpose** Display information about the prefix-based outbound route filter received from Device CE1.

**Action** From operational mode, enter the **show bgp neighbor orf detail** command.

```

user@PE1> show bgp neighbor orf 192.168.165.56 detail
Peer: 192.168.165.56 Type: External
Group: cisco-peers

inet-unicast
Filter updates rcv:          4 Immediate:          0
Filter: prefix-based         receive
  Updates rcv:              4
Received filter entries:
  seq 10 2.2.0.0/16 deny minlen 0 maxlen 0
  seq 20 3.3.0.0/16 deny minlen 24 maxlen 0
  seq 30 4.4.0.0/16 deny minlen 0 maxlen 28
  seq 40 5.5.0.0/16 deny minlen 24 maxlen 28

```

#### *Verifying the BGP Neighbor Mode*

**Purpose** Verify that the **bgp-orf-cisco-mode** setting is enabled for the peer by making sure that the **ORFCiscoMode** option is displayed in the **show bgp neighbor** command output.

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

```

user@PE1> show bgp neighbor
Peer: 192.168.165.56 AS 35          Local: 192.168.165.58 AS 65500
Type: External   State: Active     Flags: <>
Last State: Idle   Last Event: Start
Last Error: None
Export: [ adv_stat ]
Options: <Preference LocalAddress AddressFamily PeerAS Refresh>
Options: <ORF ORFCiscoMode>
Address families configured: inet-unicast
Local Address: 192.168.165.58 Holdtime: 90 Preference: 170
Number of flaps: 0
Trace options: detail open detail refresh

```

Trace file: /var/log/orf size 5242880 files 20

**Related  
Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)
- [Example: Configuring BGP to Advertise the Best External Route to Internal Peers on page 304](#)
- [Example: Setting BGP to Advertise Inactive Routes on page 298](#)

## Example: Configuring EBGP Multihop

---

- [Understanding BGP Multihop on page 241](#)
- [Example: Configuring EBGP Multihop Sessions on page 241](#)

### Understanding BGP Multihop

When external BGP (EBGP) peers are not directly connected to each other, they must cross one or more non-BGP routers to reach each other. Configuring multihop EBGP enables the peers to pass through the other routers to form peer relationships and exchange update messages. This type of configuration is typically used when a Juniper Networks routing device needs to run EBGP with a third-party router that does not allow direct connection of the two EBGP peers. EBGP multihop enables a neighbor connection between two EBGP peers that do not have a direct connection.

### Example: Configuring EBGP Multihop Sessions

This example shows how to configure an external BGP (EBGP) peer that is more than one hop away from the local router. This type of session is called a *multihop* BGP session.

- [Requirements on page 241](#)
- [Overview on page 241](#)
- [Configuration on page 242](#)
- [Verification on page 248](#)

#### Requirements

---

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

#### Overview

---

The configuration to enable multihop EBGP sessions requires connectivity between the two EBGP peers. This example uses static routes to provide connectivity between the devices.

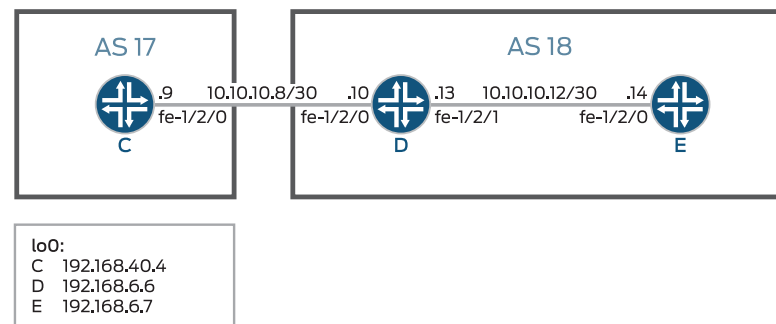
Unlike directly connected EBGP sessions in which physical address are typically used in the **neighbor** statements, you must use loopback interface addresses for multihop EBGP by specifying the loopback interface address of the indirectly connected peer. In this way, EBGP multihop is similar to internal BGP (IBGP).

Finally, you must add the **multihop** statement. Optionally, you can set a maximum time-to-live (TTL) value with the **ttl** statement. The TTL is carried in the IP header of BGP packets. If you do not specify a TTL value, the system's default maximum TTL value is used. The default TTL value is 64 for multihop EBGP sessions. Another option is to retain the BGP next-hop value for route advertisements by including the **no-nexthop-change** statement.

Figure 29 on page 242 shows a typical EBGP multihop network.

Device C and Device E have an established EBGP session. Device D is not a BGP-enabled device. All of the devices have connectivity via static routes.

**Figure 29: Typical Network with EBGP Multihop Sessions**



### 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 C**

```
set interfaces fe-1/2/0 unit 9 description to-D
set interfaces fe-1/2/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers multihop ttl 2
set protocols bgp group external-peers local-address 192.168.40.4
set protocols bgp group external-peers export send-static
set protocols bgp group external-peers peer-as 18
set protocols bgp group external-peers neighbor 192.168.6.7
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.10.10.14/32 next-hop 10.10.10.10
set routing-options static route 192.168.6.7/32 next-hop 10.10.10.10
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

**Device D**

```
set interfaces fe-1/2/0 unit 10 description to-C
set interfaces fe-1/2/0 unit 10 family inet address 10.10.10.10/30
set interfaces fe-1/2/1 unit 13 description to-E
set interfaces fe-1/2/1 unit 13 family inet address 10.10.10.13/30
set interfaces lo0 unit 4 family inet address 192.168.6.6/32
set routing-options static route 192.168.40.4/32 next-hop 10.10.10.9
```

```
set routing-options static route 192.168.6.7/32 next-hop 10.10.10.14
set routing-options router-id 192.168.6.6
```

**Device E**

```
set interfaces fe-1/2/0 unit 14 description to-D
set interfaces fe-1/2/0 unit 14 family inet address 10.10.10.14/30
set interfaces lo0 unit 5 family inet address 192.168.6.7/32
set protocols bgp group external-peers multihop ttl 2
set protocols bgp group external-peers local-address 192.168.6.7
set protocols bgp group external-peers export send-static
set protocols bgp group external-peers peer-as 17
set protocols bgp group external-peers neighbor 192.168.40.4
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.10.10.8/30 next-hop 10.10.10.13
set routing-options static route 192.168.40.4/32 next-hop 10.10.10.13
set routing-options router-id 192.168.6.7
set routing-options autonomous-system 18
```

### Device C

**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 Device C:

1. Configure the interface to the directly connected device (to-D), and configure the loopback interface.

```
[edit interfaces fe-1/2/0 unit 9]
user@C# set description to-D
user@C# set family inet address 10.10.10.9/30
```

```
[edit interfaces lo0 unit 3]
user@C# set family inet address 192.168.40.4/32
```

2. Configure an EBGP session with Device E.

The **neighbor** statement points to the loopback interface on Device E.

```
[edit protocols bgp group external-peers]
user@C# set type external
user@C# set local-address 192.168.40.4
user@C# set export send-static
user@C# set peer-as 18
user@C# set neighbor 192.168.6.7
```

3. Configure the multihop statement to enable Device C and Device E to become EBGP peers.

Because the peers are two hops away from each other, the example uses the **ttl 2** statement.

```
[edit protocols bgp group external-peers]
user@C# set multihop ttl 2
```

4. Configure connectivity to Device E, using static routes.

You must configure a route to both the loopback interface address and to the address on the physical interface.

```
[edit routing-options]
user@C# set static route 10.10.10.14/32 next-hop 10.10.10.10
user@C# set static route 192.168.6.7/32 next-hop 10.10.10.10
```

5. Configure the local router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@C# set router-id 192.168.40.4
user@C# set autonomous-system 17
```

6. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-static term 1]
user@C# set from protocol static
user@C# set then accept
```

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

```
user@C# show interfaces
fe-1/2/0 {
  unit 9 {
    description to-D;
    family inet {
      address 10.10.10.9/30;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.40.4/32;
    }
  }
}

user@C# show protocols
bgp {
  group external-peers {
    type external;
    multihop {
      ttl 2;
    }
    local-address 192.168.40.4;
    export send-static;
    peer-as 18;
    neighbor 192.168.6.7;
  }
}
```

```

user@C# show policy-options
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}

user@C# show routing-options
static {
  route 10.10.10.14/32 next-hop 10.10.10.10;
  route 192.168.6.7/32 next-hop 10.10.10.10;
}
router-id 192.168.40.4;
autonomous-system 17;

```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for all BFD sessions in the topology.

### Configuring Device D

**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 Device D:

1. Set the CLI to Device D.
2. Configure the interfaces to the directly connected devices, and configure a loopback interface.

```

[edit interfaces fe-1/2/0 unit 10]
user@D# set description to-C
user@D# set family inet address 10.10.10.10/30

```

```

[edit interfaces fe-1/2/1 unit 13]
user@D# set description to-E
user@D# set family inet address 10.10.10.13/30

```

```

[edit interfaces lo0 unit 4]
user@D# set family inet address 192.168.6.6/32

```

3. Configure connectivity to the other devices using static routes to the loopback interface addresses.

On Device D, you do not need static routes to the physical addresses because Device D is directly connected to Device C and Device E.

```

[edit routing-options]
user@D# set static route 192.168.40.4/32 next-hop 10.10.10.9
user@D# set static route 192.168.6.7/32 next-hop 10.10.10.14

```

4. Configure the local router ID.

```

[edit routing-options]

```

```
user@D# set router-id 192.168.6.6
```

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

```
user@D# show interfaces
fe-1/2/0 {
  unit 10 {
    description to-C;
    family inet {
      address 10.10.10.10/30;
    }
  }
}
fe-1/2/1 {
  unit 13 {
    description to-E;
    family inet {
      address 10.10.10.13/30;
    }
  }
}
lo0 {
  unit 4 {
    family inet {
      address 192.168.6.6/32;
    }
  }
}

user@D# show protocols

user@D# show routing-options
static {
  route 192.168.40.4/32 next-hop 10.10.10.9;
  route 192.168.6.7/32 next-hop 10.10.10.14;
}
router-id 192.168.6.6;
```

If you are done configuring the device, enter **commit** from configuration mode.  
Repeat these steps for all BFD sessions in the topology.

### **Configuring Device E**

**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 Device E:

1. Set the CLI to Device E.

```
user@host> set cli logical-system E
```

2. Configure the interface to the directly connected device (to-D), and configure the loopback interface.

```
[edit interfaces fe-1/2/0 unit 14]
user@E# set description to-D
user@E# set family inet address 10.10.10.14/30
```

```
[edit interfaces lo0 unit 5]
user@E# set family inet address 192.168.6.7/32
```

3. Configure an EBGP session with Device E.

The **neighbor** statement points to the loopback interface on Device C.

```
[edit protocols bgp group external-peers]
user@E# set local-address 192.168.6.7
user@E# set export send-static
user@E# set peer-as 17
user@E# set neighbor 192.168.40.4
```

4. Configure the **multihop** statement to enable Device C and Device E to become EBGP peers.

Because the peers are two hops away from each other, the example uses the **ttl 2** statement.

```
[edit protocols bgp group external-peers]
user@E# set multihop ttl 2
```

5. Configure connectivity to Device E, using static routes.

You must configure a route to both the loopback interface address and to the address on the physical interface.

```
[edit routing-options]
user@E# set static route 10.10.10.8/30 next-hop 10.10.10.13
user@E# set static route 192.168.40.4/32 next-hop 10.10.10.13
```

6. Configure the local router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@E# set router-id 192.168.6.7
user@E# set autonomous-system 18
```

7. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-static term 1]
user@E# set from protocol static
user@E# set then accept
```

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

```
user@E# show interfaces
```

```
fe-1/2/0 {
  unit 14 {
    description to-D;
    family inet {
      address 10.10.10.14/30;
    }
  }
}
lo0 {
  unit 5 {
    family inet {
      address 192.168.6.7/32;
    }
  }
}

user@E# show protocols
bgp {
  group external-peers {
    multihop {
      ttl 2;
    }
    local-address 192.168.6.7;
    export send-static;
    peer-as 17;
    neighbor 192.168.40.4;
  }
}

user@E# show policy-options
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}

user@E# show routing-options
static {
  route 10.10.10.8/30 next-hop 10.10.10.13;
  route 192.168.40.4/32 next-hop 10.10.10.13;
}
router-id 192.168.6.7;
autonomous-system 18;
```

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

---

## Verification

Confirm that the configuration is working properly.

- [Verifying Connectivity on page 249](#)
- [Verifying That BGP Sessions Are Established on page 249](#)
- [Viewing Advertised Routes on page 250](#)

**Verifying Connectivity**

**Purpose** Make sure that Device C can ping Device E, specifying the loopback interface address as the source of the ping request.

The loopback interface address is the source address that BGP will use.

**Action** From operational mode, enter the **ping 10.10.10.14 source 192.168.40.4** command from Device C, and enter the **ping 10.10.10.9 source 192.168.6.7** command from Device E.

```
user@C> ping 10.10.10.14 source 192.168.40.4
```

```
PING 10.10.10.14 (10.10.10.14): 56 data bytes
64 bytes from 10.10.10.14: icmp_seq=0 ttl=63 time=1.262 ms
64 bytes from 10.10.10.14: icmp_seq=1 ttl=63 time=1.202 ms
^C
--- 10.10.10.14 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.202/1.232/1.262/0.030 ms
```

```
user@E> ping 10.10.10.9 source 192.168.6.7
```

```
PING 10.10.10.9 (10.10.10.9): 56 data bytes
64 bytes from 10.10.10.9: icmp_seq=0 ttl=63 time=1.255 ms
64 bytes from 10.10.10.9: icmp_seq=1 ttl=63 time=1.158 ms
^C
--- 10.10.10.9 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.158/1.206/1.255/0.049 ms
```

**Meaning** The static routes are working if the pings work.

**Verifying That BGP Sessions Are Established**

**Purpose** Verify that the BGP sessions are up.

**Action** From operational mode, enter the **show bgp summary** command.

```
user@C> show bgp summary
```

```
Groups: 1 Peers: 1 Down peers: 0
Table      Tot Paths  Act Paths Suppressed    History  Damp State   Pending
inet.0          2         0         0         0         0         0         0
Peer          AS      InPkt   OutPkt   OutQ   Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.6.7      18      147     147       0       1    1:04:27
0/2/2/0         0/0/0/0
```

```
user@E> show bgp summary
```

```
Groups: 1 Peers: 1 Down peers: 0
Table      Tot Paths  Act Paths Suppressed    History  Damp State   Pending
inet.0          2         0         0         0         0         0         0
Peer          AS      InPkt   OutPkt   OutQ   Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.40.4     17     202     202       0       1    1:02:18
0/2/2/0         0/0/0/0
```

**Meaning** The output shows that both devices have one peer each. No peers are down.

#### Viewing Advertised Routes

**Purpose** Check to make sure that routes are being advertised by BGP.

**Action** From operational mode, enter the `show route advertising-protocol bgp neighbor` command.

```
user@C> show route advertising-protocol bgp 192.168.6.7
```

```
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref   AS path
* 10.10.10.14/32    Self
* 192.168.6.7/32    Self              I
```

```
user@E> show route advertising-protocol bgp 192.168.40.4
```

```
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref   AS path
* 10.10.10.8/30     Self
* 192.168.40.4/32   Self              I
```

**Meaning** The `send-static` routing policy is exporting the static routes from the routing table into BGP. BGP is advertising these routes between the peers because the BGP peer session is established.

- Related Documentation**
- [Examples: Configuring External BGP Peering on page 33](#)
  - [BGP Configuration Overview on page 12](#)

## Example: Configuring BGP Route Preference (Administrative Distance)

- [Understanding Route Preference Values on page 250](#)
- [Example: Configuring the Preference Value for BGP Routes on page 252](#)

### Understanding Route Preference Values

The Junos OS routing protocol process assigns a default preference value (also known as an *administrative distance*) to each route that the routing table receives. The default value depends on the source of the route. The preference value is a value from 0 through 4,294,967,295 ( $2^{32} - 1$ ), with a lower value indicating a more preferred route.

[Table 4 on page 250](#) lists the default preference values.

**Table 4: Default Route Preference Values**

| How Route Is Learned       | Default Preference | Statement to Modify Default Preference |
|----------------------------|--------------------|----------------------------------------|
| Directly connected network | 0                  | —                                      |
| System routes              | 4                  | —                                      |

Table 4: Default Route Preference Values (*continued*)

| How Route Is Learned         | Default Preference | Statement to Modify Default Preference                                                                    |
|------------------------------|--------------------|-----------------------------------------------------------------------------------------------------------|
| Static and Static LSPs       | 5                  | <i>static</i>                                                                                             |
| RSVP-signaled LSPs           | 7                  | RSVP <b>preference</b> as described in the <i>Junos OS MPLS Applications Library for Routing Devices</i>  |
| LDP-signaled LSPs            | 9                  | LDP <b>preference</b> , as described in the <i>Junos OS MPLS Applications Library for Routing Devices</i> |
| OSPF internal route          | 10                 | OSPF <i>preference</i>                                                                                    |
| IS-IS Level 1 internal route | 15                 | IS-IS <i>preference</i>                                                                                   |
| IS-IS Level 2 internal route | 18                 | IS-IS <i>preference</i>                                                                                   |
| Redirects                    | 30                 | —                                                                                                         |
| Kernel                       | 40                 | —                                                                                                         |
| SNMP                         | 50                 | —                                                                                                         |
| Router discovery             | 55                 | —                                                                                                         |
| RIP                          | 100                | RIP <i>preference</i>                                                                                     |
| RIPng                        | 100                | RIPng <i>preference</i>                                                                                   |
| PIM                          | 105                | <i>Multicast Protocols Feature Guide for Routing Devices</i>                                              |
| DVMRP                        | 110                | <i>Multicast Protocols Feature Guide for Routing Devices</i>                                              |
| Aggregate                    | 130                | <i>aggregate</i>                                                                                          |
| OSPF AS external routes      | 150                | OSPF <i>external-preference</i>                                                                           |
| IS-IS Level 1 external route | 160                | IS-IS <i>external-preference</i>                                                                          |
| IS-IS Level 2 external route | 165                | IS-IS <i>external-preference</i>                                                                          |
| BGP                          | 170                | BGP <b>preference</b> , <b>export</b> , <b>import</b>                                                     |
| MSDP                         | 175                | <i>Multicast Protocols Feature Guide for Routing Devices</i>                                              |

In general, the narrower the scope of the statement, the higher precedence its preference value is given, but the smaller the set of routes it affects. To modify the default preference value for routes learned by routing protocols, you generally apply routing policy when

configuring the individual routing protocols. You also can modify some preferences with other configuration statements, which are indicated in the table.

### Example: Configuring the Preference Value for BGP Routes

This example shows how to specify the preference for routes learned from BGP. Routing information can be learned from multiple sources. To break ties among equally specific routes learned from multiple sources, each source has a preference value. Routes that are learned through explicit administrative action, such as static routes, are preferred over routes learned from a routing protocol, such as BGP or OSPF. This concept is called *administrative distance* by some vendors.

- [Requirements on page 252](#)
- [Overview on page 252](#)
- [Configuration on page 254](#)
- [Verification on page 256](#)

---

#### Requirements

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

---

#### Overview

Routing information can be learned from multiple sources, such as through static configuration, BGP, or an interior gateway protocol (IGP). When Junos OS determines a route's preference to become the active route, it selects the route with the lowest preference as the active route and installs this route into the forwarding table. By default, the routing software assigns a preference of 170 to routes that originated from BGP. Of all the routing protocols, BGP has the highest default preference value, which means that routes learned by BGP are the least likely to become the active route.

Some vendors have a preference (distance) of 20 for external BGP (EBGP) and a distance of 200 for internal BGP (IBGP). Junos OS uses the same value (170) for both EBGP and IBGP. However, this difference between vendors has no operational impact because Junos OS always prefers EBGP routes over IBGP routes.

Another area in which vendors differ is in regard to IGP distance compared to BGP distance. For example, some vendors assign a distance of 110 to OSPF routes. This is higher than the EBGP distance of 20, and results in the selection of an EBGP route over an equivalent OSPF route. In the same scenario, Junos OS chooses the OSPF route, because of the default preference 10 for an internal OSPF route and 150 for an external OSPF route, which are both lower than the 170 preference assigned to all BGP routes.

In a multivendor environment, you might want to change the preference value for BGP routes so that Junos OS chooses an EBGP route instead of an OSPF route. To accomplish this goal, one option is to include the [preference](#) statement in the EBGP configuration. To modify the default BGP preference value, include the **preference** statement, specifying a value from 0 through 4,294,967,295 ( $2^{32} - 1$ ).



**TIP:** Another way to achieve multivendor compatibility is to include the `advertise-inactive` statement in the EBGp configuration. This causes the routing table to export to BGP the best route learned by BGP even if Junos OS did not select it to be an active route. By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. The `advertise-inactive` statement causes Junos OS to advertise the best BGP route that is inactive because of IGP preference. When you use the `advertise-inactive` statement, the Junos OS device uses the OSPF route for forwarding, and the other vendor's device uses the EBGp route for forwarding. However, from the perspective of an EBGp peer in a neighboring AS, both vendors' devices appear to behave the same way.

### Topology

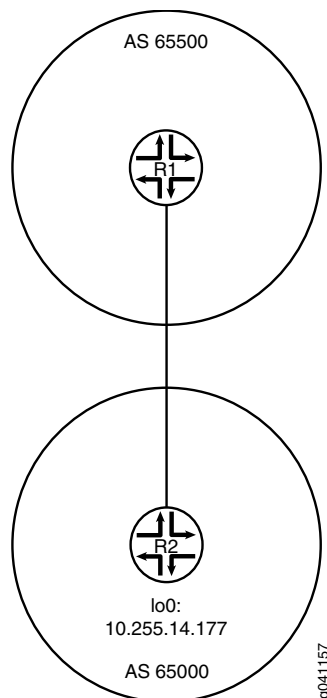
In the sample network, Device R1 and Device R2 have EBGp routes to each other and also OSPF routes to each other.

This example shows the routing tables in the following cases:

- Accept the default preference values of 170 for BGP and 10 for OSPF.
- Change the BGP preference to 8.

Figure 30 on page 253 shows the sample network.

**Figure 30: BGP Preference Value Topology**



## 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 4 family inet address 1.12.0.1/30
set interfaces lo0 unit 2 family inet address 10.255.71.24/32
set protocols bgp export send-direct
set protocols bgp group ext type external
set protocols bgp group ext preference 8
set protocols bgp group ext peer-as 65000
set protocols bgp group ext neighbor 1.12.0.2
set protocols ospf area 0.0.0.0 interface fe-1/2/0.4
set protocols ospf area 0.0.0.0 interface 10.255.71.24
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 65500
```

**Device R2**

```
set interfaces fe-1/2/0 unit 6 family inet address 1.12.0.2/30
set interfaces lo0 unit 3 family inet address 10.255.14.177/32
set protocols bgp export send-direct
set protocols bgp group ext type external
set protocols bgp group ext peer-as 65500
set protocols bgp group ext neighbor 1.12.0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6
set protocols ospf area 0.0.0.0 interface 10.255.14.177
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 65000
```

**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 Device R1:

1. Configure the interfaces.  

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 4 family inet address 1.12.0.1/30
user@R1# set lo0 unit 2 family inet address 10.255.71.24/32
```
2. Configure the local autonomous system.  

```
[edit routing-options]
user@R1# set autonomous-system 65500
```
3. Configure the external peering with Device R2.  

```
[edit protocols bgp]
user@R1# set export send-direct
user@R1# set group ext type external
user@R1# set group ext preference 8
user@R1# set group ext peer-as 65000
```

```
user@R1# set group ext neighbor 1.12.0.2
```

4. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/0.4
user@R1# set interface 10.255.71.24
```

5. Configure the routing policy.

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

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 4 {
    family inet {
      address 1.12.0.1/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 10.255.71.24/32;
    }
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show protocols
protocols {
  bgp {
    export send-direct;
    group ext {
      type external;
      preference 8;
      peer-as 65000;
      neighbor 1.12.0.2;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface fe-1/2/0.4;
    }
  }
}
```

```

        interface 10.255.71.24;
    }
}

user@R1# show routing-options
autonomous-system 65500;

```

If you are done configuring the device, enter **commit** from configuration mode.  
Repeat these steps on Device R2.

### Verification

Confirm that the configuration is working properly.

#### Verifying the Preference

**Purpose** Make sure that the routing tables on Device R1 and Device R2 reflect the fact that Device R1 is using the configured EBGp preference of 8, and Device R2 is using the default EBGp preference of 170.

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

```

user@R1> show route
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

```

1.12.0.0/30      *[Direct/0] 3d 07:03:01
                  > via fe-1/2/0.4
                  [BGP/8] 01:04:49, localpref 100
                  AS path: 65000 I
                  > to 1.12.0.2 via fe-1/2/0.4
1.12.0.1/32      *[Local/0] 3d 07:03:01
                  Local via fe-1/2/0.4
10.255.14.177/32 *[BGP/8] 01:04:49, localpref 100
                  AS path: 65000 I
                  > to 1.12.0.2 via fe-1/2/0.4
                  [OSPF/10] 3d 07:02:16, metric 1
                  > to 1.12.0.2 via fe-1/2/0.4
10.255.71.24/32  *[Direct/0] 3d 07:03:01
                  > via lo0.2
224.0.0.5/32     *[OSPF/10] 5d 03:42:16, metric 1
                  MultiRecv

```

```

user@R2> show route
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

```

1.12.0.0/30      *[Direct/0] 3d 07:03:30
                  > via fe-1/2/0.6
                  [BGP/170] 00:45:36, localpref 100
                  AS path: 65500 I
                  > to 1.12.0.1 via fe-1/2/0.6
1.12.0.2/32      *[Local/0] 3d 07:03:30
                  Local via fe-1/2/0.6
10.255.14.177/32 *[Direct/0] 3d 07:03:30
                  > via lo0.3
10.255.71.24/32  *[OSPF/10] 3d 07:02:45, metric 1
                  > to 1.12.0.1 via fe-1/2/0.6

```

```

[ BGP/170] 00:45:36, localpref 100
AS path: 65500 I
> to 1.12.0.1 via fe-1/2/0.6
224.0.0.5/32 *[OSPF/10] 5d 03:42:45, metric 1
MultiRecv

```

**Meaning** The output shows that on Device R1, the active path to Device R2's loopback interface (10.255.14.177/32) is a BGP route. The output also shows that on Device R2, the active path to Device R1's loopback interface (10.255.71.24/32) is an OSPF route.

**Related Documentation**

- [Route Preferences Overview](#)
- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Configuring BGP Path Selection

- [Understanding BGP Path Selection on page 257](#)
- [Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 260](#)

### Understanding BGP Path Selection

For each prefix in the routing table, the routing protocol process selects a single best path. After the best path is selected, the route is installed in the routing table. The best path becomes the active route if the same prefix is not learned by a protocol with a lower (more preferred) global preference value, also known as the administrative distance. The algorithm for determining the active route is as follows:

1. Verify that the next hop can be resolved.
2. Choose the path with the lowest preference value (routing protocol process preference).  
  
Routes that are not eligible to be used for forwarding (for example, because they were rejected by routing policy or because a next hop is inaccessible) have a preference of -1 and are never chosen.
3. Prefer the path with higher local preference.  
  
For non-BGP paths, choose the path with the lowest **preference2** value.
4. If the accumulated interior gateway protocol (AIGP) attribute is enabled, prefer the path with the lower AIGP attribute.
5. Prefer the path with the shortest autonomous system (AS) path value (skipped if the **as-path-ignore** statement is configured).  
  
A confederation segment (sequence or set) has a path length of 0. An AS set has a path length of 1.
6. Prefer the route with the lower origin code.

Routes learned from an IGP have a lower origin code than those learned from an exterior gateway protocol (EGP), and both have lower origin codes than incomplete routes (routes whose origin is unknown).

7. Prefer the path with the lowest multiple exit discriminator (MED) metric.

Depending on whether nondeterministic routing table path selection behavior is configured, there are two possible cases:

- If nondeterministic routing table path selection behavior is not configured (that is, if the **path-selection cisco-nondeterministic** statement is not included in the BGP configuration), for paths with the same neighboring AS numbers at the front of the AS path, prefer the path with the lowest MED metric. To always compare MEDs whether or not the peer ASs of the compared routes are the same, include the **path-selection always-compare-med** statement.
- If nondeterministic routing table path selection behavior is configured (that is, the **path-selection cisco-nondeterministic** statement is included in the BGP configuration), prefer the path with the lowest MED metric.

Confederations are not considered when determining neighboring ASs. A missing MED metric is treated as if a MED were present but zero.



**NOTE:** MED comparison works for single path selection within an AS (when the route does not include an AS path), though this usage is uncommon.

By default, only the MEDs of routes that have the same peer autonomous systems (ASs) are compared. You can configure routing table path selection options to obtain different behaviors.

8. Prefer strictly internal paths, which include IGP routes and locally generated routes (static, direct, local, and so forth).
9. Prefer strictly external BGP (EBGP) paths over external paths learned through internal BGP (IBGP) sessions.
10. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.



**NOTE:** A path is considered a BGP equal-cost path (and will be used for forwarding) if a tie-break is performed after the previous step. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor, are considered.

BGP multipath does not apply to paths that share the same MED-plus-IGP cost yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

11. If both paths are external, prefer the currently active path to minimize route-flapping. This rule is not used if any one of the following conditions is true:

- **path-selection external-router-id** is configured.
  - Both peers have the same router ID.
  - Either peer is a confederation peer.
  - Neither path is the current active path.
12. Prefer a primary route over a secondary route. A primary route is one that belongs to the routing table. A secondary route is one that is added to the routing table through an export policy.
  13. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.
  14. Prefer the path with the shortest cluster list length. The length is 0 for no list.
  15. Prefer the path from the peer with the lowest peer IP address.

### Routing Table Path Selection

The shortest AS path step of the algorithm, by default, evaluates the length of the AS path and determines the active path. You can configure an option that enables Junos OS to skip this step of the algorithm by including the **as-path-ignore** option.



**NOTE:** The **as-path-ignore** option is not supported for routing instances.

To configure routing table path selection behavior, include the **path-selection** statement:

```
path-selection {
  (always-compare-med | cisco-non-deterministic | external-router-id);
  as-path-ignore;
  med-plus-igp {
    igp-multiplier number;
    med-multiplier number;
  }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Routing table path selection can be configured in one of the following ways:

- Emulate the Cisco IOS default behavior (**cisco-non-deterministic**). This mode evaluates routes in the order that they are received and does not group them according to their neighboring AS. With **cisco-non-deterministic** mode, the active path is always first. All inactive, but eligible, paths follow the active path and are maintained in the order in which they were received, with the most recent path first. Ineligible paths remain at the end of the list.

As an example, suppose you have three path advertisements for the 192.168.1.0 /24 route:

- Path 1—learned through EBGp; AS Path of 65010; MED of 200

- Path 2—learned through IBGP; AS Path of 65020; MED of 150; IGP cost of 5
- Path 3—learned through IBGP; AS Path of 65010; MED of 100; IGP cost of 10

These advertisements are received in quick succession, within a second, in the order listed. Path 3 is received most recently, so the routing device compares it against path 2, the next most recent advertisement. The cost to the IBGP peer is better for path 2, so the routing device eliminates path 3 from contention. When comparing paths 1 and 2, the routing device prefers path 1 because it is received from an EBGP peer. This allows the routing device to install path 1 as the active path for the route.



**NOTE:** We do not recommend using this configuration option in your network. It is provided solely for interoperability to allow all routing devices in the network to make consistent route selections.

- Always comparing MEDs whether or not the peer ASs of the compared routes are the same (**always-compare-med**).
- Override the rule that If both paths are external, the currently active path is preferred (**external-router-id**). Continue with the next step (Step 12) in the path-selection process.
- Adding the IGP cost to the next-hop destination to the MED value before comparing MED values for path selection (**med-plus-igp**).

BGP multipath does not apply to paths that share the same MED-plus-IGP cost, yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

---

### Effects of Advertising Multiple Paths to a Destination

BGP advertises only the active path, unless you configure BGP to advertise multiple paths to a destination.

Suppose a routing device has in its routing table four paths to a destination and is configured to advertise up to three paths (**add-path send path-count 3**). The three paths are chosen based on path selection criteria. That is, the three best paths are chosen in path-selection order. The best path is the active path. This path is removed from consideration and a new best path is chosen. This process is repeated until the specified number of paths is reached.

### Example: Ignoring the AS Path Attribute When Selecting the Best Path

If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes of the paths. One of the route attributes that affects the best-path decision is the length of the AS paths of each route. Routes with shorter AS paths are preferred over those with longer AS paths. Although not typically practical, some scenarios might require that the AS path length be ignored in the route selection process. This example shows how to configure a routing device to ignore the AS path attribute.

- [Requirements on page 261](#)
- [Overview on page 261](#)

- [Configuration on page 262](#)
- [Verification on page 267](#)

## Requirements

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

## Overview

On externally connected routing devices, the purpose of skipping the AS path comparison might be to force an external BGP (EBGP) versus internal BGP (IBGP) decision to remove traffic from your network as soon as possible. On internally connected routing devices, you might want your IBGP-only routers to default to the local externally connected gateway. The local IBGP-only (internal) routers skip the AS path comparison and move down the decision tree to use the closest interior gateway protocol (IGP) gateway (lowest IGP metric). Doing this might be an effective way to force these routers to use a LAN connection instead of their WAN connection.



**CAUTION:** When you include the `as-path-ignore` statement on a routing device in your network, you might need to include it on all other BGP-enabled devices in your network to prevent routing loops and convergence issues. This is especially true for IBGP path comparisons.

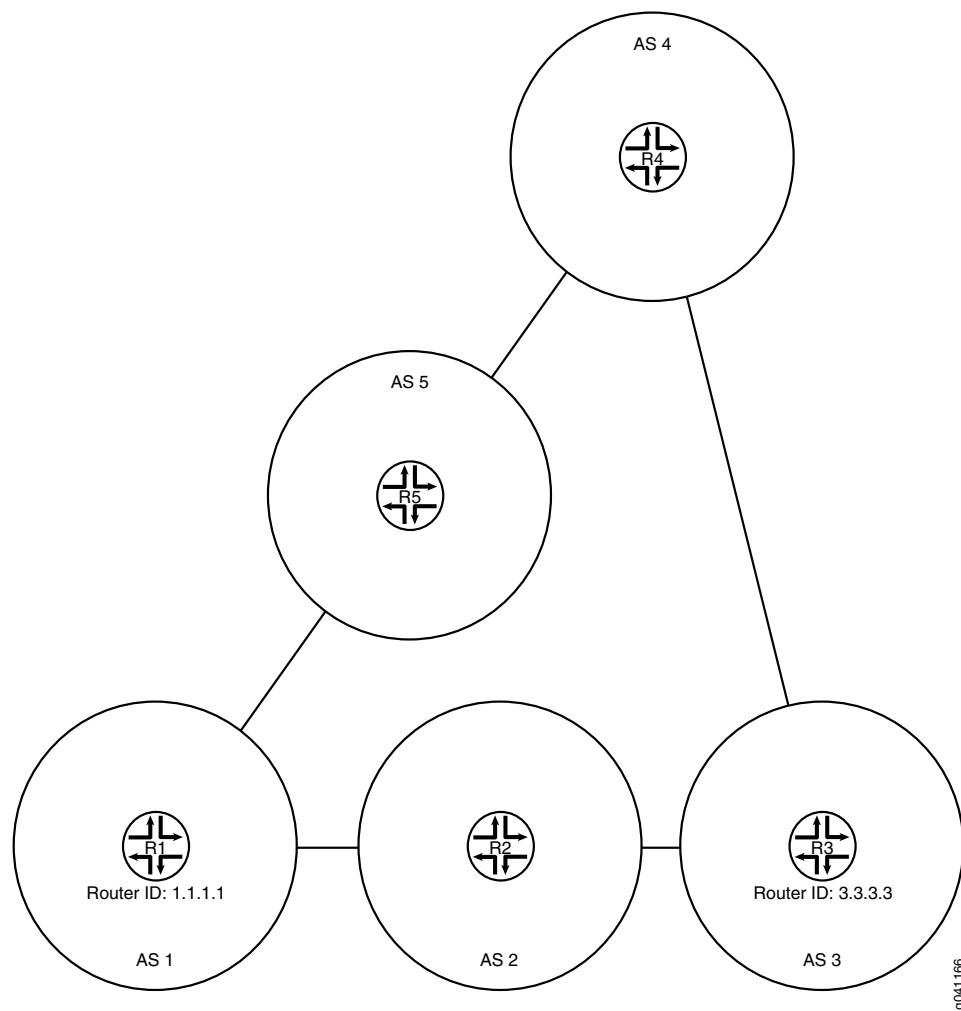
In this example, Device R2 is learning about the loopback interface address on Device R4 (4.4.4.4/32) from Device R1 and Device R3. Device R1 is advertising 4.4.4.4/32 with an AS-path of 1 5 4, and Device R3 is advertising 4.4.4.4/32 with an AS-path of 3 4. Device R2 selects the path for 4.4.4.4/32 from Device R3 as the best path because the AS path is shorter than the AS path from Device R1.

This example modifies the BGP configuration on Device R2 so that the AS-path length is not used in the best-path selection.

Device R1 has a lower router ID (1.1.1.1) than Device R3 (1.1.1.1). If all other path selection criteria are equal (or, as in this case, ignored), the route learned from Device R1 is used. Because the AS-path attribute is being ignored, the best path is toward Device R1 because of its lower router ID value.

[Figure 31 on page 262](#) shows the sample topology.

Figure 31: Topology for Ignoring the AS-Path Length



g041166

### 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 192.168.10.1/24
set interfaces fe-1/2/1 unit 10 family inet address 192.168.50.2/24
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.2 peer-as 2
set protocols bgp group ext neighbor 192.168.50.1 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
```

```

set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.50.1
set routing-options router-id 1.1.1.1
set routing-options autonomous-system 1

```

**Device R2**

```

set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.2/24
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.2/24
set interfaces lo0 unit 2 family inet address 2.2.2.2/32
set protocols bgp path-selection as-path-ignore
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.1 peer-as 1
set protocols bgp group ext neighbor 192.168.20.1 peer-as 3
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.50.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.40.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.30.0/24 next-hop 192.168.20.1
set routing-options router-id 2.2.2.2
set routing-options autonomous-system 2

```

**Device R3**

```

set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
set interfaces fe-1/2/1 unit 5 family inet address 192.168.30.1/24
set interfaces lo0 unit 3 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.20.2 peer-as 2
set protocols bgp group ext neighbor 192.168.30.2 peer-as 4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.2
set routing-options router-id 3.3.3.3
set routing-options autonomous-system 3

```

**Device R4**

```

set interfaces fe-1/2/0 unit 6 family inet address 192.168.30.2/24
set interfaces fe-1/2/1 unit 7 family inet address 192.168.40.1/24
set interfaces lo0 unit 4 family inet address 4.4.4.4/32

```

```

set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.30.1 peer-as 3
set protocols bgp group ext neighbor 192.168.40.2 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.1
set routing-options router-id 4.4.4.4
set routing-options autonomous-system 4

```

**Device R5**

```

set interfaces fe-1/2/0 unit 8 family inet address 192.168.40.2/24
set interfaces fe-1/2/1 unit 9 family inet address 192.168.50.1/24
set interfaces lo0 unit 5 family inet address 5.5.5.5/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.40.1 peer-as 4
set protocols bgp group ext neighbor 192.168.50.2 peer-as 1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.20.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.40.1
set routing-options router-id 5.5.5.5
set routing-options autonomous-system 5

```

**Configuring Device R2****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 Device R2:

1. Configure the interfaces.

```
[edit interfaces]
```

```
user@R2# set fe-1/2/0 unit 2 family inet address 192.168.10.2/24
```

```
user@R2# set fe-1/2/1 unit 3 family inet address 192.168.20.2/24
```

```
user@R2# set lo0 unit 2 family inet address 2.2.2.2/32
```

2. Configure EBGp.

```
[edit protocols bgp group ext]
```

```

user@R2# set type external
user@R2# set export send-direct
user@R2# set export send-static
user@R2# set export send-local
user@R2# set neighbor 192.168.10.1 peer-as 1
user@R2# set neighbor 192.168.20.1 peer-as 3

```

3. Configure the autonomous system (AS) path attribute to be ignored in the Junos OS path selection algorithm.

```

[edit protocols bgp]
user@R2# set path-selection as-path-ignore

```

4. Configure the routing policy.

```

[edit policy-options]
user@R2# set policy-statement send-direct term 1 from protocol direct
user@R2# set policy-statement send-direct term 1 then accept
user@R2# set policy-statement send-local term 1 from protocol local
user@R2# set policy-statement send-local term 1 then accept
user@R2# set policy-statement send-static term 1 from protocol static
user@R2# set policy-statement send-static term 1 then accept

```

5. Configure some static routes.

```

[edit routing-options static]
user@R2# set route 192.168.50.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.40.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.30.0/24 next-hop 192.168.20.1

```

6. Configure the autonomous system (AS) number and the router ID.

```

[edit routing-options]
user@R2# set router-id 2.2.2.2
user@R2# set autonomous-system 2

```

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

```

user@R2# show interfaces
fe-1/2/0 {
  unit 2 {
    family inet {
      address 192.168.10.2/24;
    }
  }
}
fe-1/2/1 {
  unit 3 {
    family inet {
      address 192.168.20.2/24;
    }
  }
}
lo0 {
  unit 2 {

```

```
        family inet {
            address 2.2.2.2/32;
        }
    }
}

user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
policy-statement send-local {
    term 1 {
        from protocol local;
        then accept;
    }
}
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}

user@R2# show protocols
bgp {
    path-selection as-path-ignore;
    group ext {
        type external;
        export [ send-direct send-static send-local ];
        neighbor 192.168.10.1 {
            peer-as 1;
        }
        neighbor 192.168.20.1 {
            peer-as 3;
        }
    }
}

user@R2# show routing-options
static {
    route 192.168.50.0/24 next-hop 192.168.10.1;
    route 192.168.40.0/24 next-hop 192.168.10.1;
    route 192.168.30.0/24 next-hop 192.168.20.1;
}
router-id 2.2.2.2;
autonomous-system 2;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on the other devices in the network, changing the interface names and IP addresses, as needed.

## Verification

Confirm that the configuration is working properly.

- [Checking the Neighbor Status on page 267](#)

### Checking the Neighbor Status

**Purpose** Make sure that from Device R2, the active path to get to AS 4 is through AS 1 and AS 5, not through AS 3.



**NOTE:** To verify the functionality of the `as-path-ignore` statement, you might need to run the `restart routing` command to force reevaluation of the active path. This is because for BGP, if both paths are external, the Junos OS behavior is to prefer the currently active path. This behavior helps to minimize route-flapping. Use caution when restarting the routing protocol process in a production network.

**Action** From operational mode, enter the `restart routing` command.

```
user@R2> restart routing
Routing protocols process started, pid 49396
```

From operational mode, enter the `show route 4.4.4.4 protocol bgp` command.

```
user@R2> show route 4.4.4.4 protocol bgp
inet.0: 12 destinations, 25 routes (12 active, 0 holddown, 4 hidden)
+ = Active Route, - = Last Active, * = Both

4.4.4.4/32          *[BGP/170] 00:00:12, localpref 100
                    AS path: 1 5 4 I
                    > to 192.168.10.1 via fe-1/2/0.2
                    [BGP/170] 00:00:08, localpref 100
                    AS path: 3 4 I
                    > to 192.168.20.1 via fe-1/2/1.3
```

**Meaning** The asterisk (\*) is next to the path learned from R1, meaning that this is the active path. The AS path for the active path is 1 5 4, which is longer than the AS path (3 4) for the nonactive path learned from Router R3.

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Removing Private AS Numbers

- [Understanding Private AS Number Removal from AS Paths on page 268](#)
- [Example: Removing Private AS Numbers from AS Paths on page 269](#)

## Understanding Private AS Number Removal from AS Paths

By default, when BGP advertises AS paths to remote systems, it includes all AS numbers, including private AS numbers. You can configure the software so that it removes private AS numbers from AS paths. Doing this is useful when any of the following circumstances are true:

- A remote AS for which you provide connectivity is multihomed, but only to the local AS.
- The remote AS does not have an officially allocated AS number.
- It is not appropriate to make the remote AS a confederation member AS of the local AS.

Most companies acquire their own AS number. Some companies also use private AS numbers to connect to their public AS network. These companies might use a different private AS number for each region in which their company does business. In any implementation, announcing a private AS number to the Internet must be avoided. Service providers can use the **remove-private** statement to prevent advertising private AS numbers to the Internet.

In an enterprise scenario, suppose that you have multiple AS numbers in your company, some of which are private AS numbers, and one with a public AS number. The one with a public AS number has a direct connection to the service provider. In the AS that connects directly to the service provider, you can use the **remove-private** statement to filter out any private AS numbers in the advertisements that are sent to the service provider.



**CAUTION:** Changing configuration statements that affect BGP peers, such as enabling or disabling **remove-private** or renaming a BGP group, resets the BGP sessions. Changes that affect BGP peers should only be made when resetting a BGP session is acceptable.

---

The AS numbers are stripped from the AS path starting at the left end of the AS path (the end where AS paths have been most recently added). The routing device stops searching for private ASs when it finds the first nonprivate AS or a peer's private AS. If the AS path contains the AS number of the external BGP (EBGP) neighbor, BGP does not remove the private AS number.

---



**NOTE:** As of Junos OS 10.0R2 and later, if there is a need to send prefixes to an EBGP peer that has an AS number that matches an AS number in the AS path, consider using the **as-override** statement instead of the **remove-private** statement.

---

The operation takes place after any confederation member ASs have already been removed from the AS path, if applicable.

The software is preconfigured with knowledge of the set of AS numbers that is considered private, a range that is defined in the Internet Assigned Numbers Authority (IANA) assigned numbers document. The set of AS numbers reserved as private are in the range from 64,512 through 65,534, inclusive.

### Example: Removing Private AS Numbers from AS Paths

This example demonstrates the removal of a private AS number from the advertised AS path to avoid announcing the private AS number to the Internet.

- [Requirements on page 269](#)
- [Overview on page 269](#)
- [Configuration on page 270](#)
- [Verification on page 272](#)

#### Requirements

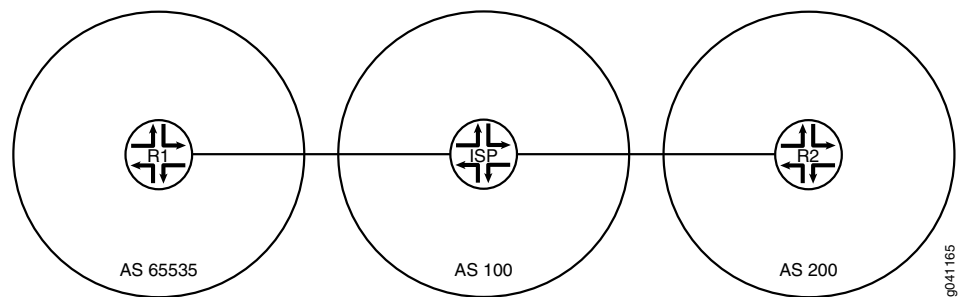
No special configuration beyond device initialization is required before you configure this example.

#### Overview

Service providers and enterprise networks use the **remove-private** statement to prevent advertising private AS numbers to the Internet. The **remove-private** statement works in the outbound direction. You configure the **remove-private** statement on a device that has a public AS number and that is connected to one or more devices that have private AS numbers. Generally, you would not configure this statement on a device that has a private AS number.

[Figure 32 on page 269](#) shows the sample topology.

**Figure 32: Topology for Removing a Private AS from the Advertised AS Path**



In this example, Device R1 is connected to its service provider using private AS number 65535. The example shows the **remove-private** statement configured on Device ISP to prevent Device R1's private AS number from being announced to Device R2. Device R2 sees only the AS number of the service provider.

## 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 192.168.10.1/24
set interfaces lo0 unit 1 family inet address 10.10.10.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 192.168.10.10
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.10
set routing-options autonomous-system 65535
```

**Device ISP**

```
set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.10/24
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.20/24
set interfaces lo0 unit 2 family inet address 10.10.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext neighbor 192.168.10.1 peer-as 65535
set protocols bgp group ext neighbor 192.168.20.1 remove-private
set protocols bgp group ext neighbor 192.168.20.1 peer-as 200
set routing-options autonomous-system 100
```

**Device R2**

```
set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
set interfaces lo0 unit 3 family inet address 10.10.20.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 192.168.20.20
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.20.20
set routing-options autonomous-system 200
```

### Device ISP

**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 Device ISP:

1. Configure the interfaces.

[edit interfaces]

```

user@ISP# set fe-1/2/0 unit 2 family inet address 192.168.10.10/24
user@ISP# set fe-1/2/1 unit 3 family inet address 192.168.20.20/24
user@ISP# set lo0 unit 2 family inet address 10.10.0.1/32

```

2. Configure EBGP.

```

[edit protocols bgp group ext]
user@ISP# set type external
user@ISP# set neighbor 192.168.10.1 peer-as 65535
user@ISP# set neighbor 192.168.20.1 peer-as 200

```

3. For the neighbor in autonomous system (AS) 200 (Device R2), remove private AS numbers from the advertised AS paths.

```

[edit protocols bgp group ext]
user@ISP# set neighbor 192.168.20.1 remove-private

```

4. Configure the AS number.

```

[edit routing-options]
user@ISP# set autonomous-system 100

```

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

```

user@ISP# show interfaces
fe-1/2/0 {
  unit 2 {
    family inet {
      address 192.168.10.10/24;
    }
  }
}
fe-1/2/1 {
  unit 3 {
    family inet {
      address 192.168.20.20/24;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 10.10.0.1/32;
    }
  }
}
}

user@ISP# show protocols
bgp {
  group ext {
    type external;
    neighbor 192.168.10.1 {
      peer-as 65535;
    }
    neighbor 192.168.20.1 {
      remove-private;
    }
  }
}

```

```

        peer-as 200;
    }
}

user@ISP# show routing-options
autonomous-system 100;

```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on Device R1 and Device R2, changing the interface names and IP address, as needed, and adding the routing policy configuration.

### Verification

Confirm that the configuration is working properly.

- [Checking the Neighbor Status on page 272](#)
- [Checking the Routing Tables on page 273](#)
- [Checking the AS Path When the remove-private Statement Is Deactivated on page 273](#)

#### Checking the Neighbor Status

**Purpose** Make sure that Device ISP has the **remove-private** setting enabled in its neighbor session with Device R2.

**Action** From operational mode, enter the **show bgp neighbor 192.168.20.1** command.

```

user@ISP> show bgp neighbor 192.168.20.1
Peer: 192.168.20.1+179 AS 200 Local: 192.168.20.20+60216 AS 100
  Type: External  State: Established  Flags: <ImportEval Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference RemovePrivateAS PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.20.1      Local ID: 10.10.0.1      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/1.3
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 200)
  Peer does not support Addpath
  Table inet.0 Bit: 10001
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          1
    Received prefixes:        3
    Accepted prefixes:        2
    Suppressed due to damping: 0
    Advertised prefixes:      1

```

```

Last traffic (seconds): Received 10   Sent 16   Checked 55
Input messages: Total 54   Updates 3   Refreshes 0   Octets 1091
Output messages: Total 54   Updates 1   Refreshes 0   Octets 1118
Output Queue[0]: 0

```

**Meaning** The `RemovePrivateAS` option shows that Device ISP has the expected setting.

### *Checking the Routing Tables*

**Purpose** Make sure that the devices have the expected routes and AS paths.

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

```

user@R1> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.20.1/32      *[BGP/170] 00:28:57, localpref 100
                  AS path: 100 200 I
                  > to 192.168.10.10 via fe-1/2/0.1

user@ISP> show route protocol bgp

inet.0: 7 destinations, 11 routes (7 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:29:40, localpref 100
                  AS path: 65535 I
                  > to 192.168.10.1 via fe-1/2/0.2
10.10.20.1/32      *[BGP/170] 00:29:36, localpref 100
                  AS path: 200 I
                  > to 192.168.20.1 via fe-1/2/1.3
192.168.10.0/24    [BGP/170] 00:29:40, localpref 100
                  AS path: 65535 I
                  > to 192.168.10.1 via fe-1/2/0.2
192.168.20.0/24    [BGP/170] 00:29:36, localpref 100
                  AS path: 200 I
                  > to 192.168.20.1 via fe-1/2/1.3

user@R2> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:29:53, localpref 100
                  AS path: 100 I
                  > to 192.168.20.20 via fe-1/2/0.4

```

**Meaning** Device ISP has the private AS number 65535 in its AS path to Device R1. However, Device ISP does not advertise this private AS number to Device R2. This is shown in the routing table of Device R2. Device R2's path to Device R1 contains only the AS number for Device ISP.

### *Checking the AS Path When the remove-private Statement Is Deactivated*

**Purpose** Verify that without the `remove-private` statement, the private AS number appears in Device R2's routing table.

**Action** From configuration mode on Device ISP, enter the **deactivate remove-private** command and then recheck the routing table on Device R2.

```
[protocols bgp group ext neighbor 192.168.20.1]
user@ISP# deactivate remove-private
user@ISP# commit

user@R2> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:00:54, localpref 100
                  AS path: 100 65535 I
                  > to 192.168.20.20 via fe-1/2/0.4
```

**Meaning** Private AS number 65535 appears in Device R2's AS path to Device R1.

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

---

## Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers

---

- [Understanding the Default BGP Routing Policy on Packet Transport Routers on page 274](#)
- [Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers on page 276](#)

### Understanding the Default BGP Routing Policy on Packet Transport Routers

On PTX Series Packet Transport Routers, the default BGP routing policy differs from that of other Junos OS routing devices.

The PTX Series routers are MPLS transit platforms that do IP forwarding, typically using interior gateway protocol (IGP) routes. The PTX Series Packet Forwarding Engine can accommodate a relatively small number of variable-length prefixes.



**NOTE:** A PTX Series router can support full BGP routes in the control plane so that it can be used as a route reflector (RR). It can do exact-length lookup multicast forwarding and can build the multicast forwarding plane for use by the unicast control plane (for example, to perform a reverse-path forwarding lookup for multicast).

---

Given the PFE limitation, the default routing policy for PTX Series routers is for BGP routes not to be installed in the forwarding table. You can override the default routing policy and select certain BGP routes to install in the forwarding table.

The default behavior for load balancing and BGP routes on PTX Series routers is as follows. It has the following desirable characteristics:

- Allows you to override the default behavior without needing to alter the default policy directly
- Reduces the chance of accidental changes that nullify the defaults
- Sets no flow-control actions, such as accept and reject

The default routing policy on the PTX Series routers is as follows:

```
user@host# show policy-options | display inheritance defaults no-comments
policy-options {
  policy-statement junos-ptx-series-default {
    term t1 {
      from {
        protocol bgp;
        rib inet.0;
      }
      then no-install-to-fib;
    }
    term t2 {
      from {
        protocol bgp;
        rib inet6.0;
      }
      then no-install-to-fib;
    }
    term t3 {
      then load-balance per-packet;
    }
  }
}
routing-options {
  forwarding-table {
    default-export junos-ptx-series-default;
  }
}
user@host# show routing-options forwarding-table default-export | display inheritance
defaults no-comments
default-export junos-ptx-series-default;
```

As shown here, the **junos-ptx-series-default** policy is defined in **[edit policy-options]**. The policy is applied in **[edit routing-options forwarding-table]**, using the **default-export** statement. You can view these default configurations by using the **| display inheritance** flag.

Also, you can use the **show policy** command to view the default policy.

```
user@host> show policy junos-ptx-series-default
Policy junos-ptx-series-default:
  Term t1:
    from proto BGP
    inet.0
    then install-to-fib no
  Term t2:
```

```

from proto BGP
  inet6.0
  then install-to-fib no
Term t3:
  then load-balance per-packet

```



**CAUTION:** We strongly recommend that you do not alter the `junos-ptx-series-default` routing policy directly.

Junos OS chains the `junos-ptx-series-default` policy and any user-configured export policy. Because the `junos-ptx-series-default` policy does not use flow-control actions, any export policy that you configure is executed (by way of the implicit next-policy action) for every route. Thus you can override any actions set by the `junos-ptx-series-default` policy. If you do not configure an export policy, the actions set by `junos-ptx-series-default` policy are the only actions.

You can use the policy action `install-to-fib` to override the `no-install-to-fib` action.

Similarly, you can set the `load-balance per-prefix` action to override the `load-balance per-packet` action.

## Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers

This example shows how to override the default routing policy on packet transport routers, such as the PTX Series Packet Transport Routers.

- [Requirements on page 276](#)
- [Overview on page 276](#)
- [Configuration on page 277](#)
- [Verification on page 278](#)

### Requirements

This example requires Junos OS Release 12.1 or later.

### Overview

By default, the PTX Series routers do not install BGP routes in the forwarding table.

For PTX Series routers, the configuration of the `from protocols bgp` condition with the `then accept` action does not have the usual result that it has on other Junos OS routing devices. With the following routing policy on PTX Series routers, BGP routes do not get installed in the forwarding table.

```

user@host# show policy-options
policy-statement accept-no-install {
  term 1 {
    from protocol bgp;
    then accept;
  }
}
user@host# show routing-options

```

```

forwarding-table {
    export accept-no-install;
}

user@host> show route forwarding-table
Routing table: default.inet
Internet:
Destination          Type RtRef Next hop          Type Index NhRef Netif
default              perm  0                rjct   36    2

```

No BGP routes are installed in the forwarding table. This is the expected behavior.

This example shows how to use the **then install-to-fib** action to effectively override the default BGP routing policy.

### 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.

```

set policy-options prefix-list install-bgp 66.0.0.1/32
set policy-options policy-statement override-ptx-series-default term 1 from prefix-list
install-bgp
set policy-options policy-statement override-ptx-series-default term 1 then load-balance
per-prefix
set policy-options policy-statement override-ptx-series-default term 1 then install-to-fib
set routing-options forwarding-table export override-ptx-series-default

```

### Installing Selected BGP Routes in the Forwarding Table

**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 install selected BGP routes in the forwarding table:

1. Configure a list of prefixes to install in the forwarding table.
 

```

[edit policy-options prefix-list install-bgp]
user@host# set 66.0.0.1/32

```
2. Configure the routing policy, applying the prefix list as a condition.
 

```

[edit policy-options policy-statement override-ptx-series-default term 1]
user@host# set from prefix-list install-bgp
user@host# set then install-to-fib
user@host# set then load-balance per-prefix

```
3. Apply the routing policy to the forwarding table.
 

```

[edit routing-options forwarding-table]
user@host# set export override-ptx-series-default

```

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

```

user@host# show policy-options
prefix-list install-bgp {
  66.0.0.1/32;
}
policy-statement override-ptx-series-default {
  term 1 {
    from {
      prefix-list install-bgp;
    }
    then {
      load-balance per-prefix;
      install-to-fib;
    }
  }
}

user@host# show routing-options
forwarding-table {
  export override-ptx-series-default;
}

```

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

### Verification

Confirm that the configuration is working properly.

#### *Verifying That the Selected Route Is Installed in the Forwarding Table*

**Purpose** Make sure that the configured policy overrides the default policy.

**Action** From operational mode, enter the [show route forwarding-table](#) command.

```

user@host> show route forwarding-table destination 66.0.0.1
Internet:
Destination          Type RtRef Next hop          Type Index NhRef Netif
66.0.0.1/32          user    0              5.1.0.2          ucst   574    1 et-6/0/0.1
                    5.2.0.2          ucst   575    1 et-6/0/0.2

```

**Meaning** This output shows that the route to 66.0.0.1/32 is installed in the forwarding table.

**Related Documentation**

- [Default Routing Policies](#)
- [Examples: Configuring BGP Multipath on page 365](#)

## Example: Configuring Conditional Installation of Prefixes in a Routing Table

- [Conditional Installation of Prefixes Use Cases on page 279](#)
- [Understanding Conditional Installation of Prefixes in a Routing Table on page 281](#)
- [Example: Configuring Conditional Installation of Prefixes in a Routing Table on page 282](#)

## Conditional Installation of Prefixes Use Cases

Networks are usually subdivided into smaller, more-manageable units called autonomous systems (ASs). When BGP is used by routers to form peer relationships in the same AS, it is referred to as internal BGP (IBGP). When BGP is used by routers to form peer relationships in different ASs, it is referred to as external BGP (EBGP).

After performing route sanity checks, a BGP router accepts the routes received from its peers and installs them into the routing table. By default, all routers in IBGP and EBGP sessions follow the standard BGP advertisement rules. While a router in an IBGP session advertises only the routes learned from its direct peers, a router in an EBGP session advertises all routes learned from its direct and indirect peers (peers of peers). Hence, in a typical network configured with EBGP, a router adds all routes received from an EBGP peer into its routing table and advertises nearly all routes to all EBGP peers.

A service provider exchanging BGP routes with both customers and peers on the Internet is at risk of malicious and unintended threats that can compromise the proper routing of traffic, as well as the operation of the routers.

This has several disadvantages:

- **Non-aggregated route advertisements**—A customer could erroneously advertise all its prefixes to the ISP rather than an aggregate of its address space. Given the size of the Internet routing table, this must be carefully controlled. An edge router might also need only a default route out toward the Internet and instead be receiving the entire BGP routing table from its upstream peer.
- **BGP route manipulation**—If a malicious administrator alters the contents of the BGP routing table, it could prevent traffic from reaching its intended destination.
- **BGP route hijacking**—A rogue administrator of a BGP peer could maliciously announce a network's prefixes in an attempt to reroute the traffic intended for the victim network to the administrator's network to either gain access to the contents of traffic or to block the victim's online services.
- **BGP denial of service (DoS)**—If a malicious administrator sends unexpected or undesirable BGP traffic to a router in an attempt to use all of the router's available BGP resources, it might result in impairing the router's ability to process valid BGP route information.

Conditional installation of prefixes can be used to address all the problems previously mentioned. If a customer requires access to remote networks, it is possible to install a specific route in the routing table of the router that is connected with the remote network. This does not happen in a typical EBGP network and hence, conditional installation of prefixes becomes essential.

ASs are not only bound by physical relationships but by business or other organizational relationships. An AS can provide services to another organization, or act as a transit AS between two other ASs. These transit ASs are bound by contractual agreements between the parties that include parameters on how to connect to each other and most importantly, the type and quantity of traffic they carry for each other. Therefore, for both legal and financial reasons, service providers must implement policies that control how

BGP routes are exchanged with neighbors, which routes are accepted from those neighbors, and how those routes affect the traffic between the ASs.

There are many different options available to filter routes received from a BGP peer to both enforce inter-AS policies and mitigate the risks of receiving potentially harmful routes. Conventional route filtering examines the attributes of a route and accepts or rejects the route based on such attributes. A policy or filter can examine the contents of the AS-Path, the next-hop value, a community value, a list of prefixes, the address family of the route, and so on.

In some cases, the standard “acceptance condition” of matching a particular attribute value is not enough. The service provider might need to use another condition outside of the route itself, for example, another route in the routing table. As an example, it might be desirable to install a default route received from an upstream peer, only if it can be verified that this peer has reachability to other networks further upstream. This conditional route installation avoids installing a default route that is used to send traffic toward this peer, when the peer might have lost its routes upstream, leading to black-holed traffic. To achieve this, the router can be configured to search for the presence of a particular route in the routing table, and based on this knowledge accept or reject another prefix.

[“Example: Configuring Conditional Installation of Prefixes in a Routing Table” on page 282](#) explains how the conditional installation of prefixes can be configured and verified.

## Understanding Conditional Installation of Prefixes in a Routing Table

BGP accepts all non-looped routes learned from neighbors and imports them into the RIB-In table. If these routes are accepted by the BGP import policy, they are then imported into the inet.0 routing table. In cases where only certain routes are required to be imported, provisions can be made such that the peer routing device exports routes based on a condition or a set of conditions.

The condition for exporting a route can be based on:

- The peer the route was learned from
- The interface the route was learned on
- Some other required attribute

For example:

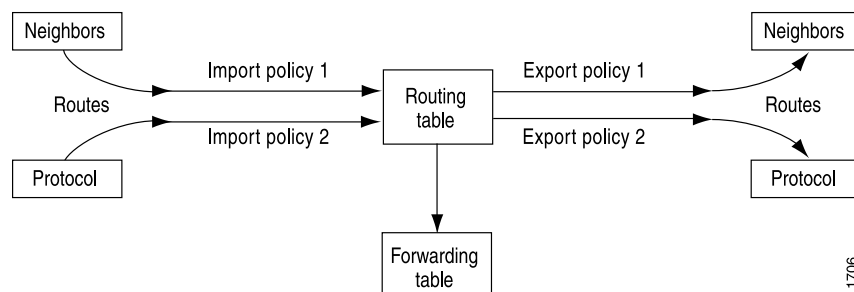
```
[edit]
policy-options {
  condition condition-name {
    if-route-exists address table table-name;
  }
}
```

This is known as conditional installation of prefixes and is described in [“Example: Configuring Conditional Installation of Prefixes in a Routing Table”](#) on page 282.

The Juniper Networks® Junos® Operating System (Junos OS) supports conditional export of routes based on the existence of another route in the routing table. Junos OS does not, however, support policy conditions for import policy.

[Figure 33 on page 281](#) illustrates where BGP import and export policies are applied. An import policy is applied to inbound routes that are visible in the output of the **show route receive-protocol bgp neighbor-address** command. An export policy is applied to outbound routes that are visible in the output of the **show route advertising-protocol bgp neighbor-address** command.

**Figure 33: BGP Import and Export Policies**



To enable conditional installation of prefixes, an export policy must be configured on the device where the prefix export has to take place. The export policy evaluates each route to verify that it satisfies all the match conditions under the **from** statement. It also searches

for the existence of the route defined under the **condition** statement (also configured under the **from** statement).

If the route does not match the entire set of required conditions defined in the policy, or if the route defined under the **condition** statement does not exist in the routing table, the route is not exported to its BGP peers. Thus, a conditional export policy matches the routes for the desired route or prefix you want installed in the peers' routing table.

To configure the conditional installation of prefixes with the help of an export policy:

1. Create a **condition** statement to check prefixes.

```
[edit]
policy-options {
  condition condition-name {
    if-route-exists address table table-name;
  }
}
```

2. Create an export policy with the newly created condition using the **condition** statement.

```
[edit]
policy-options {
  policy-statement policy-name {
    term 1 {
      from {
        protocols bgp;
        condition condition-name;
      }
      then {
        accept;
      }
    }
  }
}
```

3. Apply the export policy to the device that requires only selected prefixes to be exported from the routing table.

```
[edit]
protocols bgp {
  group group-name {
    export policy-name;
  }
}
```

## Example: Configuring Conditional Installation of Prefixes in a Routing Table

This example shows how to configure conditional installation of prefixes in a routing table using BGP export policy.

- [Requirements on page 283](#)
- [Overview on page 283](#)
- [Configuration on page 285](#)
- [Verification on page 292](#)

## Requirements

This example uses the following hardware and software components:

- M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers
- Junos OS Release 9.0 or later

## Overview

In this example, three routers in three different autonomous systems (ASs) are connected and configured with the BGP protocol. Router Internet, which is the upstream router, has five addresses configured on its lo0.0 loopback interface (11.1.1.1/32, 12.1.1.1/32, 13.1.1.1, 14.1.1.1/32, and 15.1.1.1/32), and an extra loopback address (192.168.9.1/32) to be configured as the router ID. These six addresses are exported into BGP to emulate the contents of a BGP routing table of a router connected to the Internet, and advertised to Router North.

Router North exports a default route into BGP, and advertises the default route and the five BGP routes to Router South, which is the downstream router. Router South receives the default route and only one other route (11.1.1.1/32), and installs this route and the default route in its routing table.

To summarize, the example meets the following requirements:

- On Device North, send 0/0 to Device South only if a particular route is also sent (in the example 11.1.1.1/32).
- On Device South, accept the default route and the 11.1.1.1/32 route. Drop all other routes. Consider that Device South might be receiving the entire Internet table, while the operator only wants Device South to have the default and one other specific prefix.

The first requirement is met with an export policy on Device North:

```
user@North# show policy-options
policy-statement conditional-export-bgp {
  term prefix_11 {
    from {
      protocol bgp;
      route-filter 11.0.0.0/5 orlonger;
    }
    then accept;
  }
  term conditional-default {
    from {
      route-filter 0.0.0.0/0 exact;
      condition prefix_11;
    }
    then accept;
  }
  term others {
    then reject;
  }
}
```

```
condition prefix_11 {  
  if-route-exists {  
    11.1.1.1/32;  
    table inet.0;  
  }  
}
```

The logic of the conditional export policy can be summarized as follows: If 0/0 is present, and if 11.1.1.1/32 is present, then send the 0/0 prefix. This implies that if 11.1.1.1/32 is not present, then do not send 0/0.

The second requirement is met with an import policy on Device South:

```
user@South# show policy-options  
policy-statement import-selected-routes {  
  term 1 {  
    from {  
      rib inet.0;  
      neighbor 10.0.78.14;  
      route-filter 0.0.0.0/0 exact;  
      route-filter 11.0.0.0/8 orlonger;  
    }  
    then accept;  
  }  
  term 2 {  
    then reject;  
  }  
}
```

In this example, four routes are dropped as a result of the import policy on Device South. This is because the export policy on Device North leaks all of the routes received from Device Internet, and the import policy on Device South excludes some of these routes.

It is important to understand that in Junos OS, although an import policy (inbound route filter) might reject a route, not use it for traffic forwarding, and not include it in an advertisement to other peers, the router retains these routes as hidden routes. These hidden routes are not available for policy or routing purposes. However, they do occupy memory space on the router. A service provider filtering routes to control the amount of information being kept in memory and processed by a router might want the router to entirely drop the routes being rejected by the import policy.

Hidden routes can be viewed by using the **show route receive-protocol bgp neighbor-address hidden** command. The hidden routes can then be retained or dropped from the routing table by configuring the **keep all | none** statement at the **[edit protocols bgp]** or **[edit protocols bgp group group-name]** hierarchy level.

The rules of BGP route retention are as follows:

- By default, all routes learned from BGP are retained, except those where the AS path is looped. (The AS path includes the local AS.)
- By configuring the **keep all** statement, all routes learned from BGP are retained, even those with the local AS in the AS path.

- By configuring the **keep none** statement, BGP discards routes that were received from a peer and that were rejected by import policy or other sanity checking. When this statement is configured and the inbound policy changes, Junos OS re-advertises all the routes advertised by the peer.

When you configure **keep all** or **keep none** and the peers support route refresh, the local speaker sends a refresh message and performs an import evaluation. For these peers, the sessions do not restart. To determine if a peer supports refresh, check for **Peer supports Refresh capability** in the output of the **show bgp neighbor** command.

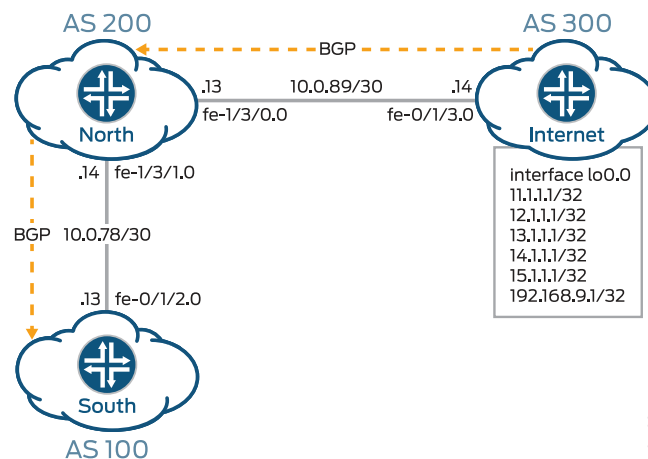


**CAUTION:** If you configure **keep all** or **keep none** and the peer does not support session restart, the associated BGP sessions are restarted (flapped).

### Topology

Figure 34 on page 285 shows the topology used in this example.

Figure 34: Conditional Installation of Prefixes



### 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.

#### Router Internet

```
set interfaces lo0 unit 0 family inet address 11.1.1.1/32
set interfaces lo0 unit 0 family inet address 12.1.1.1/32
set interfaces lo0 unit 0 family inet address 13.1.1.1/32
set interfaces lo0 unit 0 family inet address 14.1.1.1/32
set interfaces lo0 unit 0 family inet address 15.1.1.1/32
set interfaces lo0 unit 0 family inet address 192.168.9.1/32
set interfaces fe-0/1/3 unit 0 family inet address 10.0.89.14/30
set protocols bgp group toNorth local-address 10.0.89.14
set protocols bgp group toNorth peer-as 200
set protocols bgp group toNorth neighbor 10.0.89.13
set protocols bgp group toNorth export into-bgp
```

```
set policy-options policy-statement into-bgp term 1 from interface lo0.0
set policy-options policy-statement into-bgp term 1 then accept
set routing-options router-id 192.168.9.1
set routing-options autonomous-system 300
```

**Router North**

```
set interfaces fe-1/3/1 unit 0 family inet address 10.0.78.14/30
set interfaces fe-1/3/0 unit 0 family inet address 10.0.89.13/30
set interfaces lo0 unit 0 family inet address 192.168.8.1/32
set protocols bgp group toInternet local-address 10.0.89.13
set protocols bgp group toInternet peer-as 300
set protocols bgp group toInternet neighbor 10.0.89.14
set protocols bgp group toSouth local-address 10.0.78.14
set protocols bgp group toSouth export conditional-export-bgp
set protocols bgp group toSouth peer-as 100
set protocols bgp group toSouth neighbor 10.0.78.13
set policy-options policy-statement conditional-export-bgp term prefix_11 from protocol
    bgp
set policy-options policy-statement conditional-export-bgp term prefix_11 from route-filter
    11.0.0.0/5 orlonger
set policy-options policy-statement conditional-export-bgp term prefix_11 then accept
set policy-options policy-statement conditional-export-bgp term conditional-default
    from route-filter 0.0.0.0/0 exact
set policy-options policy-statement conditional-export-bgp term conditional-default
    from condition prefix_11
set policy-options policy-statement conditional-export-bgp term conditional-default
    then accept
set policy-options policy-statement conditional-export-bgp term others then reject
set policy-options condition prefix_11 if-route-exists 11.1.1.1/32
set policy-options condition prefix_11 if-route-exists table inet.0
set routing-options static route 0/0 reject
set routing-options router-id 192.168.8.1
set routing-options autonomous-system 200
```

**Router South**

```
set interfaces fe-0/1/2 unit 0 family inet address 10.0.78.13/30
set interfaces lo0 unit 0 family inet address 192.168.7.1/32
set protocols bgp group toNorth local-address 10.0.78.13
set protocols bgp group toNorth import import-selected-routes
set protocols bgp group toNorth peer-as 200
set protocols bgp group toNorth neighbor 10.0.78.14
set policy-options policy-statement import-selected-routes term 1 from neighbor 10.0.78.14
set policy-options policy-statement import-selected-routes term 1 from route-filter
    11.0.0.0/8 orlonger
set policy-options policy-statement import-selected-routes term 1 from route-filter
    0.0.0.0/0 exact
set policy-options policy-statement import-selected-routes term 1 then accept
set policy-options policy-statement import-selected-routes term 2 then reject
set routing-options router-id 192.168.7.1
set routing-options autonomous-system 100
```

### Configuring Conditional Installation of Prefixes

**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 Junos OS CLI User Guide.

To configure conditional installation of prefixes:

1. Configure the router interfaces forming the links between the three routers.

```
Router Internet
[edit interfaces]
user@Internet# set fe-0/1/3 unit 0 family inet address 10.0.89.14/30
```

```
Router North
[edit interfaces]
user@North# set fe-1/3/1 unit 0 family inet address 10.0.78.14/30
user@North# set fe-1/3/0 unit 0 family inet address 10.0.89.13/30
```

```
Router South
[edit interfaces]
user@South# set fe-0/1/2 unit 0 family inet address 10.0.78.13/30
```

2. Configure five loopback interface addresses on Router Internet to emulate BGP routes learned from the Internet that are to be imported into the routing table of Router South, and configure an additional address (192.168.9.1/32) that will be configured as the router ID.

```
Router Internet
[edit interfaces lo0 unit 0 family inet]
user@Internet# set address 11.1.1.1/32
user@Internet# set address 12.1.1.1/32
user@Internet# set address 13.1.1.1/32
user@Internet# set address 14.1.1.1/32
user@Internet# set address 15.1.1.1/32
user@Internet# set address 192.168.9.1/32
```

Also, configure the loopback interface addresses on Routers North and South.

```
Router North
[edit interfaces lo0 unit 0 family inet]
user@North# set address 192.168.8.1/32
```

```
Router South
[edit interfaces lo0 unit 0 family inet]
user@South# set address 192.168.7.1/32
```

3. Configure the static default route on Router North to be advertised to Router South.

```
[edit routing-options]
user@North# set static route 0/0 reject
```

4. Define the condition for exporting prefixes from the routing table on Router North.

```
[edit policy-options condition prefix_11]
user@North# set if-route-exists 11.1.1.1/32
user@North# set if-route-exists table inet.0
```

5. Define export policies (**into-bgp** and **conditional-export-bgp**) on Routers Internet and North respectively, to advertise routes to BGP.



**NOTE:** Ensure that you reference the condition, `prefix_11` (configured in Step 4), in the export policy.

#### Router Internet

```
[edit policy-options policy-statement into-bgp ]
user@Internet# set term 1 from interface lo0.0
user@Internet# set term 1 then accept
```

#### Router North

```
[edit policy-options policy-statement conditional-export-bgp]
user@North# set term prefix_11 from protocol bgp
user@North# set term prefix_11 from route-filter 11.0.0.0/5 orlonger
user@North# set term prefix_11 then accept
user@North# set term conditional-default from route-filter 0.0.0.0/0 exact
user@North# set term conditional-default from condition prefix_11
user@North# set term conditional-default then accept
user@North# set term others then reject
```

6. Define an import policy (**import-selected-routes**) on Router South to import some of the routes advertised by Router North into its routing table.

```
[edit policy-options policy-statement import-selected-routes ]
user@South# set term 1 from neighbor 10.0.78.14
user@South# set term 1 from route-filter 11.0.0.0/8 orlonger
user@South# set term 1 from route-filter 0.0.0.0/0 exact
user@South# set term 1 then accept
user@South# set term 2 then reject
```

7. Configure BGP on all three routers to enable the flow of prefixes between the autonomous systems.



**NOTE:** Ensure that you apply the defined import and export policies to the respective BGP groups for prefix advertisement to take place.

#### Router Internet

```
[edit protocols bgp group toNorth]
user@Internet# set local-address 10.0.89.14
user@Internet# set peer-as 200
user@Internet# set neighbor 10.0.89.13
user@Internet# set export into-bgp
```

#### Router North

```
[edit protocols bgp group toInternet]
user@North# set local-address 10.0.89.13
user@North# set peer-as 300
user@North# set neighbor 10.0.89.14
```

```
[edit protocols bgp group toSouth]
user@North# set local-address 10.0.78.14
user@North# set peer-as 100
user@North# set neighbor 10.0.78.13
user@North# set export conditional-export-bgp
```

**Router South**

```
[edit protocols bgp group toNorth]
user@South# set local-address 10.0.78.13
user@South# set peer-as 200
user@South# set neighbor 10.0.78.14
user@South# set import import-selected-routes
```

8. Configure the router ID and autonomous system number for all three routers.



**NOTE:** In this example, the router ID is configured based on the IP address configured on the lo0.0 interface of the router.

**Router Internet**

```
[edit routing options]
user@Internet# set router-id 192.168.9.1
user@Internet# set autonomous-system 300
```

**Router North**

```
[edit routing options]
user@North# set router-id 192.168.8.1
user@North# set autonomous-system 200
```

**Router South**

```
[edit routing options]
user@South# set router-id 192.168.7.1
user@South# set autonomous-system 100
```

**Results**

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

```
Device Internet user@Internet# show interfaces
fe-0/1/3 {
  unit 0 {
    family inet {
      address 10.0.89.14/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 11.1.1.1/32;
      address 12.1.1.1/32;
      address 13.1.1.1/32;
      address 14.1.1.1/32;
      address 15.1.1.1/32;
      address 192.168.9.1/32;
    }
  }
}
```

```
user@Internet# show protocols bgp
group toNorth {
  local-address 10.0.89.14;
  export into-bgp;
  peer-as 200;
  neighbor 10.0.89.13;
}

user@Internet# show policy-options
policy-statement into-bgp {
  term 1 {
    from interface lo0.3;
    then accept;
  }
}

user@Internet# show routing-options
router-id 192.168.9.1;
autonomous-system 300;

Device North user@North# show interfaces
fe-1/3/1 {
  unit 0 {
    family inet {
      address 10.0.78.14/30;
    }
  }
}
fe-1/3/0 {
  unit 0 {
    family inet {
      address 10.0.89.13/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.8.1/32;
    }
  }
}

user@North# show protocols bgp
group toInternet {
  local-address 10.0.89.13;
  peer-as 300;
  neighbor 10.0.89.14;
}
group toSouth {
  local-address 10.0.78.14;
  export conditional-export-bgp;
  peer-as 100;
  neighbor 10.0.78.13;
}

user@North# show policy-options
```

```

policy-statement conditional-export-bgp {
  term prefix_11 {
    from {
      protocol bgp;
      route-filter 11.0.0.0/5 orlonger;
    }
    then accept;
  }
  term conditional-default {
    from {
      route-filter 0.0.0.0/0 exact;
      condition prefix_11;
    }
    then accept;
  }
  term others {
    then reject;
  }
}
condition prefix_11 {
  if-route-exists {
    11.1.1.1/32;
    table inet.0;
  }
}

user@North# show routing-options
static {
  route 0.0.0.0/0 reject;
}
router-id 192.168.8.1;
autonomous-system 200;

Device South user@South# show interfaces
fe-0/1/2 {
  unit 0 {
    family inet {
      address 10.0.78.13/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.7.1/32;
    }
  }
}

user@South# show protocols bgp
bgp {
  group toNorth {
    local-address 10.0.78.13;
    import import-selected-routes;
    peer-as 200;
    neighbor 10.0.78.14;
  }
}

```

```

    }
  }

user@South# show policy-options
policy-statement import-selected-routes {
  term 1 {
    from {
      neighbor 10.0.78.14;
      route-filter 11.0.0.0/8 orlonger;
      route-filter 0.0.0.0/0 exact;
    }
    then accept;
  }
  term 2 {
    then reject;
  }
}

user@South# show routing-options
router-id 192.168.7.1;
autonomous-system 100;

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying BGP on page 292](#)
- [Verifying Prefix Advertisement from Router Internet to Router North on page 294](#)
- [Verifying Prefix Advertisement from Router North to Router South on page 295](#)
- [Verifying BGP Import Policy for Installation of Prefixes on page 295](#)
- [Verifying Conditional Export from Router North to Router South on page 296](#)
- [Verifying the Presence of Routes Hidden by Policy \(Optional\) on page 296](#)

### Verifying BGP

**Purpose** Verify that BGP sessions have been established between the three routers.

**Action** From operational mode, run the **show bgp neighbor *neighbor-address*** command.

1. Check the BGP session on Router Internet to verify that Router North is a neighbor.

```

user@Internet> show bgp neighbor 10.0.89.13
Peer: 10.0.89.13+179 AS 200 Local: 10.0.89.14+56187 AS 300
  Type: External   State: Established   Flags: [ImportEval Sync]
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ into-bgp ]
  Options: [Preference LocalAddress PeerAS Refresh]
  Local Address: 10.0.89.14 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.8.1      Local ID: 192.168.9.1      Active Holdtime: 90
  Keepalive Interval: 30    Group index: 0      Peer index: 0
  BFD: disabled, down

```

```

Local Interface: fe-0/1/3.0
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Stale routes from peer are kept for: 300
Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 200)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      6
Last traffic (seconds): Received 9   Sent 18   Checked 28
Input messages: Total 12   Updates 1   Refreshes 0   Octets 232
Output messages: Total 14   Updates 1   Refreshes 0   Octets 383
Output Queue[0]: 0

```

2. Check the BGP session on Router North to verify that Router Internet is a neighbor.

```

user@North> show bgp neighbor 10.0.89.14
Peer: 10.0.89.14+56187 AS 300 Local: 10.0.89.13+179 AS 200
  Type: External   State: Established   Flags: [ImportEval Sync]
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: [Preference LocalAddress PeerAS Refresh]
  Local Address: 10.0.89.13 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.9.1      Local ID: 192.168.8.1      Active Holdtime: 90
  Keepalive Interval: 30    Group index: 0      Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/3/0.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 300)
  Peer does not support Addpath
  Table inet.0 Bit: 10001
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          6
    Received prefixes:        6
    Accepted prefixes:        6
    Suppressed due to damping: 0
    Advertised prefixes:      0
  Last traffic (seconds): Received 14   Sent 3   Checked 3
  Input messages: Total 16   Updates 2   Refreshes 0   Octets 402

```

```
Output messages: Total 15      Updates 0      Refreshes 0      Octets 348
Output Queue[0]: 0
```

Check the following fields in these outputs to verify that BGP sessions have been established:

- **Peer**—Check if the peer AS number is listed.
- **Local**—Check if the local AS number is listed.
- **State**—Ensure that the value is **Established**. If not, check the configuration again and see [show bgp neighbor](#) for more details on the output fields.

Similarly, verify that Routers North and South form peer relationships with each other.

**Meaning** BGP sessions are established between the three routers.

#### *Verifying Prefix Advertisement from Router Internet to Router North*

**Purpose** Verify that the routes sent from Router Internet are received by Router North.

- Action** 1. From operational mode on Router Internet, run the **show route advertising-protocol bgp 10.0.89.13** command.

```
user@Internet> show route advertising-protocol bgp 10.0.89.13
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref  AS path
* 11.1.1.1/32           Self              I
* 12.1.1.1/32           Self              I
* 13.1.1.1/32           Self              I
* 14.1.1.1/32           Self              I
* 15.1.1.1/32           Self              I
* 192.168.9.1/32        Self              I
```

The output verifies that Router Internet advertises the routes 11.1.1.1/32, 12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, 15.1.1.1/32, and 192.168.9.1/32 (the loopback address used as router ID) to Router North.

2. From operational mode on Router North, run the **show route receive-protocol bgp neighbor-address** command.

```
user@North> show route receive-protocol bgp 10.0.89.14
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref  AS path
* 11.1.1.1/32           10.0.89.14       300 I
* 12.1.1.1/32           10.0.89.14       300 I
* 13.1.1.1/32           10.0.89.14       300 I
* 14.1.1.1/32           10.0.89.14       300 I
* 15.1.1.1/32           10.0.89.14       300 I
* 192.168.9.1/32        10.0.89.14       300 I
```

The output verifies that Router North has received all the routes advertised by Router Internet.

**Meaning** Prefixes sent by Router Internet have been successfully installed into the routing table on Router North.

**Verifying Prefix Advertisement from Router North to Router South**

**Purpose** Verify that the routes received from Router Internet and the static default route are advertised by Router North to Router South.

**Action** 1. From operational mode on Router North, run the **show route 0/0 exact** command.

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

```
0.0.0.0/0          *[Static/5] 00:10:22
                   Reject
```

The output verifies the presence of the static default route (0.0.0.0/0) in the routing table on Router North.

2. From operational mode on Router North, run the **show route advertising-protocol bgp neighbor-address** command.

```
user@North> show route advertising-protocol bgp 10.0.78.13
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref  AS path
* 0.0.0.0/0             Self                      I
* 11.1.1.1/32           Self                      300 I
* 12.1.1.1/32           Self                      300 I
* 13.1.1.1/32           Self                      300 I
* 14.1.1.1/32           Self                      300 I
* 15.1.1.1/32           Self                      300 I
```

The output verifies that Router North is advertising the static route and the 11.1.1.1/32 route received from Router Internet, as well as many other routes, to Router South.

**Verifying BGP Import Policy for Installation of Prefixes**

**Purpose** Verify that the BGP import policy successfully installs the required prefixes.

**Action** See if the import policy on Router South is operational by checking if only the static default route from Router North and the 11.1.1.1/32 route from Router South are installed in the routing table.

From operational mode, run the **show route receive-protocol bgp neighbor-address** command.

```
user@South> show route receive-protocol bgp 10.0.78.14
inet.0: 10 destinations, 11 routes (6 active, 0 holddown, 4 hidden)
  Prefix                Nexthop          MED      Lclpref  AS path
* 0.0.0.0/0             10.0.78.14      200 I
* 11.1.1.1/32           10.0.78.14      200 300 I
```

The output verifies that the BGP import policy is operational on Router South, and only the static default route of 0.0.0.0/0 from Router North and the 11.1.1.1/32 route from Router Internet have leaked into the routing table on Router South.

**Meaning** The installation of prefixes is successful because of the configured BGP import policy.

**Verifying Conditional Export from Router North to Router South**

**Purpose** Verify that when Device Internet stops sending the 11.1.1.1/32 route, Device North stops sending the default 0/0 route.

- Action** 1. Cause Device Internet to stop sending the 11.1.1.1/32 route by deactivating the 11.1.1.1/32 address on the loopback interface.

```
[edit interfaces lo0 unit 0 family inet]
user@Internet# deactivate address 11.1.1.1/32
user@Internet# commit
```

2. From operational mode on Router North, run the **show route advertising-protocol bgp neighbor-address** command.

```
user@North> show route advertising-protocol bgp 10.0.78.13
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref  AS path
* 12.1.1.1/32           Self              300      I
* 13.1.1.1/32           Self              300      I
* 14.1.1.1/32           Self              300      I
* 15.1.1.1/32           Self              300      I
```

The output verifies that Router North is not advertising the default route to Router South. This is the expected behavior when the 11.1.1.1/32 route is not present.

3. Reactivate the 11.1.1.1/32 address on Device Internet's loopback interface.

```
[edit interfaces lo0 unit 0 family inet]
user@Internet# activate address 11.1.1.1/32
user@Internet# commit
```

**Verifying the Presence of Routes Hidden by Policy (Optional)**

**Purpose** Verify the presence of routes hidden by the import policy configured on Router South.



**NOTE:** This section demonstrates the effects of various changes you can make to the configuration depending on your needs.

**Action** View routes hidden from the routing table of Router South by:

- Using the **hidden** option for the **show route receive-protocol bgp neighbor-address** command.
- Deactivating the import policy.

1. From operational mode, run the **show route receive-protocol bgp neighbor-address hidden** command to view hidden routes.

```
user@South> show route receive-protocol bgp 10.0.78.14 hidden
inet.0: 10 destinations, 11 routes (6 active, 0 holddown, 4 hidden)
  Prefix                Nexthop          MED      Lc1pref    AS path
  12.1.1.1/32           10.0.78.14             200 300 I
  13.1.1.1/32           10.0.78.14             200 300 I
  14.1.1.1/32           10.0.78.14             200 300 I
  15.1.1.1/32           10.0.78.14             200 300 I
```

The output verifies the presence of routes hidden by the import policy (12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, and 15.1.1.1/32) on Router South.

2. Deactivate the BGP import policy by configuring the **deactivate import** statement at the **[edit protocols bgp group group-name]** hierarchy level.

```
[edit protocols bgp group toNorth]
user@South# deactivate import
user@South# commit
```

3. Run the **show route receive-protocol bgp neighbor-address** operational mode command to check the routes after deactivating the import policy.

```
user@South> show route receive-protocol bgp 10.0.78.14
inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref    AS path
* 0.0.0.0/0             10.0.78.14             200 I
* 11.1.1.1/32           10.0.78.14             200 300 I
* 12.1.1.1/32           10.0.78.14             200 300 I
* 13.1.1.1/32           10.0.78.14             200 300 I
* 14.1.1.1/32           10.0.78.14             200 300 I
* 15.1.1.1/32           10.0.78.14             200 300 I
```

The output verifies the presence of previously hidden routes (12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, and 15.1.1.1/32).

4. Activate the BGP import policy and remove the hidden routes from the routing table by configuring the **activate import** and **keep none** statements respectively at the **[edit protocols bgp group group-name]** hierarchy level.

```
[edit protocols bgp group toNorth]
user@South# activate import
user@South# set keep none
user@South# commit
```

5. From operational mode, run the **show route receive-protocol bgp neighbor-address hidden** command to check the routes after activating the import policy and configuring the **keep none** statement.

```
user@South> show route receive-protocol bgp 10.0.78.14 hidden

inet.0: 6 destinations, 7 routes (6 active, 0 holddown, 0 hidden)
```

The output verifies that the hidden routes are not maintained in the routing table because of the configured **keep none** statement.

## Example: Setting BGP to Advertise Inactive Routes

---

By default, BGP readvertises only active routes. To have the routing table export to BGP the best route learned by BGP even if Junos OS did not select it to be an active route, include the **advertise-inactive** statement:

```
advertise-inactive;
```

In Junos OS, BGP advertises BGP routes that are installed or active, which are routes selected as the best based on the BGP path selection rules. The **advertise-inactive** statement allows nonactive BGP routes to be advertised to other peers.

Junos OS also provides support for configuring a BGP export policy that matches the state of an advertised route. You can match either active or inactive routes, as follows:

```
policy-options {  
  policy-statement name{  
    from state (active|inactive);  
  }  
}
```

This qualifier only matches when used in the context of an export policy. When a route is being advertised by a protocol that can advertise inactive routes (such as BGP), **state inactive** matches routes advertised as a result of the **advertise-inactive** (or **advertise-external**) statement.

For example, the following configuration can be used as a BGP export policy to mark routes advertised due to the **advertise-inactive** setting with a user-defined community. That community can be later used by the receiving routers to filter out such routes from the forwarding table. Such a mechanism can be used to address concerns that advertising paths not used for forwarding by the sender might lead to forwarding loops.

```
user@host# show policy-options  
policy-statement mark-inactive {  
  term inactive {  
    from state inactive;  
    then {  
      community set comm-inactive;  
    }  
  }  
  term default {  
    from protocol bgp;  
    then accept;  
  }  
  then reject;  
}  
community comm-inactive members 65535:65284;
```

- [Requirements on page 299](#)
- [Overview on page 299](#)
- [Configuration on page 299](#)
- [Verification on page 302](#)

## Requirements

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

## Overview

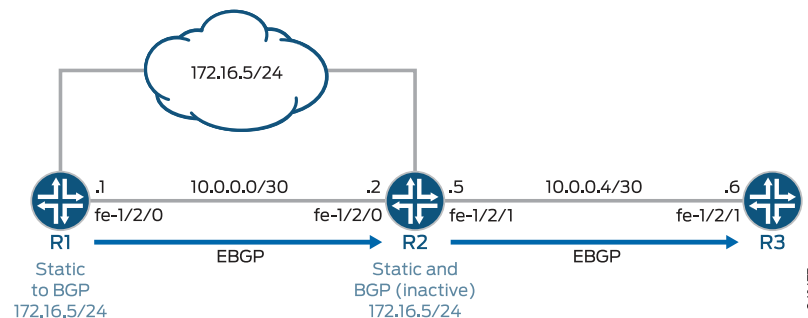
In this example, Device R2 has two external BGP (EBGP) peers, Device R1 and Device R3.

Device R1 has a static route to 172.16.5/24. Likewise, Device R2 also has a static route to 172.16.5/24. Through BGP, Device R1 sends information about its static route to Device R2. Device R2 now has information about 172.16.5/24 from two sources—its own static route and the BGP-learned route received from Device R1. Static routes are preferred over BGP-learned routes, so the BGP route is inactive on Device R2. Normally Device R2 would send the BGP-learned information to Device R3, but Device R2 does not do this because the BGP route is inactive. Device R3, therefore, has no information about 172.16.5/24 unless you enable the **advertise-inactive** command on Device R2, which causes Device R2 to send the BGP-learned to Device R3.

## Topology

Figure 35 on page 299 shows the sample network.

Figure 35: BGP Topology for advertise-inactive



“CLI Quick Configuration” on page 299 shows the configuration for all of the devices in Figure 35 on page 299.

The section “Step-by-Step Procedure” on page 300 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 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group to_R2 type external
set protocols bgp group to_R2 export send-static
set protocols bgp group to_R2 neighbor 10.0.0.2 peer-as 200
set policy-options policy-statement send-static term 1 from protocol static
```

```
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 172.16.5.0/24 discard
set routing-options static route 172.16.5.0/24 install
set routing-options autonomous-system 100
```

**Device R2**

```
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 0 family inet address 10.0.0.5/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group to_R1 type external
set protocols bgp group to_R1 neighbor 10.0.0.1 peer-as 100
set protocols bgp group to_R3 type external
set protocols bgp group to_R3 advertise-inactive
set protocols bgp group to_R3 neighbor 10.0.0.6 peer-as 300
set routing-options static route 172.16.5.0/24 discard
set routing-options static route 172.16.5.0/24 install
set routing-options autonomous-system 200
```

**Device R3**

```
set interfaces fe-1/2/1 unit 0 family inet address 10.0.0.6/30
set interfaces fe-1/2/0 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.5
set routing-options autonomous-system 300
```

**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 Device R2:

1. Configure the device interfaces.

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

user@R2# set fe-1/2/1 unit 0 family inet address 10.0.0.5/30

user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure the EBGP connection to Device R1.

```
[edit protocols bgp group to_R1]
user@R2# set type external
user@R2# set neighbor 10.0.0.1 peer-as 100
```

3. Configure the EBGP connection to Device R3.

```
[edit protocols bgp group to_R3]
user@R2# set type external
user@R2# set neighbor 10.0.0.6 peer-as 300
```

4. Add the **advertise-inactive** statement to the EBGP group peering session with Device R3.

```
[edit protocols bgp group to_R3]
user@R2# set advertise-inactive
```

5. Configure the static route to the 172.16.5.0/24 network.

```
[edit routing-options static]
user@R2# set route 172.16.5.0/24 discard
user@R2# set route 172.16.5.0/24 install
```

6. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    family inet {
      address 10.0.0.5/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.2/32;
    }
  }
}
}

user@R2# show protocols
bgp {
  group to_R1 {
    type external;
    neighbor 10.0.0.1 {
      peer-as 100;
    }
  }
  group to_R3 {
    type external;
    advertise-inactive;
    neighbor 10.0.0.6 {
      peer-as 300;
    }
  }
}

user@R2# show routing-options
```

```
static {
  route 172.16.5.0/24 {
    discard;
    install;
  }
}
autonomous-system 200;
```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying the BGP Active Path on page 302](#)
- [Verifying the External Route Advertisement on page 302](#)
- [Verifying the Route on Device R3 on page 303](#)
- [Experimenting with the advertise-inactive Statement on page 303](#)

### Verifying the BGP Active Path

**Purpose** On Device R2, make sure that the 172.16.5.0/24 prefix is in the routing table and has the expected active path.

**Action** user@R2> show route 172.16.5

```
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.5.0/24      *[Static/5] 21:24:38
                   Discard
                   [BGP/170] 21:21:41, localpref 100
                   AS path: 100 I, validation-state: unverified
                   > to 10.0.0.1 via fe-1/2/0.0
```

**Meaning** Device R2 receives the 172.16.5.0/24 route from both Device R1 and from its own statically configured route. The static route is the active path, as designated by the asterisk (\*). The static route path has the lowest route preference (5) as compared to the BGP preference (170). Therefore, the static route becomes active.

### Verifying the External Route Advertisement

**Purpose** On Device R2, make sure that the 172.16.5.0/24 route is advertised toward Device R3.

**Action** user@R2> show route advertising-protocol bgp 10.0.0.6

```
inet.0: 6 destinations, 7 routes (6 active, 0 holddown, 0 hidden)
Prefix                Nexthop            MED      Lclpref  AS path
172.16.5.0/24         Self              0         0         100 I
```

**Meaning** Device R2 is advertising the 172.16.5.0/24 route toward Device R3

### Verifying the Route on Device R3

- Purpose** Make sure that the 172.16.6.0/24 prefix is in Device R3's routing table.
- Action** `user@R3> show route 172.16.5.0/24`
- ```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.5.0/24      *[BGP/170] 00:01:19, localpref 100
                   AS path: 200 100 I, validation-state: unverified
                   > to 10.0.0.5 via fe-1/2/1.0
```
- Meaning** Device R3 has the BGP-learned route for 172.16.5.0/24.

### Experimenting with the advertise-inactive Statement

- Purpose** See what happens when the **advertise-inactive** statement is removed from the BGP configuration on Device R2.
- Action**
- On Device R2, deactivate the **advertise-inactive** statement.
 

```
[edit protocols bgp group to_R3]
user@R2# deactivate advertise-inactive
user@R2# commit
```
  - On Device R2, check to see if the 172.16.5.0/24 route is advertised toward Device R3.
 

```
user@R2> show route advertising-protocol bgp 10.0.0.6
```

As expected, the route is no longer advertised.
  - On Device R3, ensure that the 172.16.5.0/24 route is absent from the routing table.
 

```
user@R3> show route 172.16.5/24
```
- Meaning** Device R1 advertises route 172.16.5.0/24 to Device R2, but Device R2 has a manually configured static route for this prefix. Static routes are preferred over BGP routes, so Device R2 installs the BGP route as an inactive route. Because the BGP route is not active, Device R2 does not readvertise the BGP route to Device R3. This is the default behavior in Junos OS. If you add the **advertise-inactive** statement to the BGP configuration on Device R2, Device R2 readvertises nonactive routes.
- Related Documentation**
- [Example: Configuring BGP to Advertise the Best External Route to Internal Peers on page 304](#)
  - [Understanding BGP Path Selection on page 8](#)

## Example: Configuring BGP to Advertise the Best External Route to Internal Peers

---

The BGP protocol specification, as defined in RFC 1771, specifies that a BGP peer shall advertise to its internal peers the higher preference external path, even if this path is not the overall best (in other words, even if the best path is an internal path). In practice, deployed BGP implementations do not follow this rule. The reasons for deviating from the specification are as follows:

- Minimizing the amount of advertised information. BGP scales according to the number of available paths.
- Avoiding routing and forwarding loops.

There are, however, several scenarios in which the behavior, specified in RFC 1771, of advertising the best external route might be beneficial. Limiting path information is not always desirable as path diversity might help reduce restoration times. Advertising the best external path can also address internal BGP (IBGP) route oscillation issues as described in RFC 3345, *Border Gateway Protocol (BGP) Persistent Route Oscillation Condition*.

The **advertise-external** statement modifies the behavior of a BGP speaker to advertise the best external path to IBGP peers, even when the best overall path is an internal path.



**NOTE:** The **advertise-external** statement is supported at both the group and neighbor level. If you configure the statement at the neighbor level, you must configure it for all neighbors in a group. Otherwise, the group is automatically split into different groups.

The **conditional** option limits the behavior of the **advertise-external** setting, such that the external route is advertised only if the route selection process reaches the point where the multiple exit discriminator (MED) metric is evaluated. Thus, an external route is not advertised if it has, for instance, an AS path that is worse (longer) than that of the active path. The **conditional** option restricts external path advertisement to when the best external path and the active path are equal until the MED step of the route selection process. Note that the criteria used for selecting the best external path is the same whether or not the **conditional** option is configured.

Junos OS also provides support for configuring a BGP export policy that matches the state of an advertised route. You can match either active or inactive routes, as follows:

```
policy-options {  
  policy-statement name{  
    from state (active|inactive);  
  }  
}
```

This qualifier only matches when used in the context of an export policy. When a route is being advertised by a protocol that can advertise inactive routes (such as BGP), **state**

**inactive** matches routes advertised as a result of the **advertise-inactive** and **advertise-external** statements.

For example, the following configuration can be used as a BGP export policy toward internal peers to mark routes advertised due to the **advertise-external** setting with a user-defined community. That community can be later used by the receiving routers to filter out such routes from the forwarding table. Such a mechanism can be used to address concerns that advertising paths not used for forwarding by the sender might lead to forwarding loops.

```
user@host# show policy-options
policy-statement mark-inactive {
  term inactive {
    from state inactive;
    then {
      community set comm-inactive;
    }
  }
  term default {
    from protocol bgp;
    then accept;
  }
  then reject;
}
community comm-inactive members 65535:65284;
```

- [Requirements on page 305](#)
- [Overview on page 305](#)
- [Configuration on page 306](#)
- [Verification on page 309](#)

## Requirements

Junos OS 9.3 or later is required.

## Overview

This example shows three routing devices. Device R2 has an external BGP (EBGP) connection to Device R1. Device R2 has an IBGP connection to Device R3.

Device R1 advertises 172.16.6.0/24. Device R2 does not set the local preference in an import policy for Device R1's routes, and thus 172.16.6.0/24 has the default local preference of 100.

Device R3 advertises 172.16.6.0/24 with a local preference of 200.

When the **advertise-external** statement is not configured on Device R2, 172.16.6.0/24 is not advertised by Device R2 toward Device R3.

When the **advertise-external** statement is configured on Device R2 on the session toward Device R3, 172.16.6.0/24 is advertised by Device R2 toward Device R3.

When **advertise-external conditional** is configured on Device R2 on the session toward Device R3, 172.16.6.0/24 is not advertised by Device R2 toward Device R3. If you remove

the **then local-preference 200** setting on Device R3 and add the **path-selection as-path-ignore** setting on Device R2 (thus making the path selection criteria equal until the MED step of the route selection process), 172.16.6.0/24 is advertised by Device R2 toward Device R3.



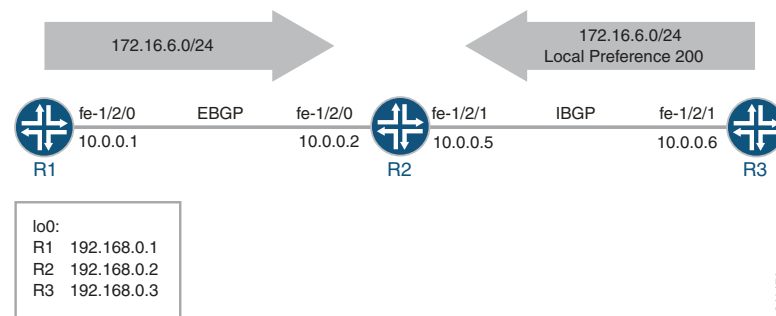
**NOTE:** To configure the **advertise-external** statement on a route reflector, you must disable intracluster reflection with the **no-client-reflect** statement, and the client cluster must be fully meshed to prevent the sending of redundant route advertisements.

When a routing device is configured as a route reflector for a cluster, a route advertised by the route reflector is considered internal if it is received from an internal peer with the same cluster identifier or if both peers have no cluster identifier configured. A route received from an internal peer that belongs to another cluster, that is, with a different cluster identifier, is considered external.

## Topology

Figure 36 on page 306 shows the sample network.

Figure 36: BGP Topology for advertise-external



"CLI Quick Configuration" on page 306 shows the configuration for all of the devices in Figure 36 on page 306.

The section "Step-by-Step Procedure" on page 307 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 0 description to-R2
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 200
```

```

set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 from route-filter 172.16.6.0/24
  exact
set policy-options policy-statement send-static term 1 then accept
set policy-options policy-statement send-static term 2 then reject
set routing-options static route 172.16.6.0/24 reject
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 100

```

**Device R2**

```

set interfaces fe-1/2/0 unit 0 description to-R1
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 0 description to-R3
set interfaces fe-1/2/1 unit 0 family inet address 10.0.0.5/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group ext type external
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 10.0.0.1
set protocols bgp group int type internal
set protocols bgp group int local-address 192.168.0.2
set protocols bgp group int advertise-external
set protocols bgp group int neighbor 192.168.0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 200

```

**Device R3**

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp group int type internal
set protocols bgp group int local-address 192.168.0.3
set protocols bgp group int export send-static
set protocols bgp group int neighbor 192.168.0.2
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then local-preference 200
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 172.16.6.0/24 reject
set routing-options static route 0.0.0.0/0 next-hop 10.0.0.5
set routing-options autonomous-system 200

```

**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 Device R2:

1. Configure the device interfaces.

[edit interfaces]

user@R2# set fe-1/2/0 unit 0 description to-R1

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

user@R2# set fe-1/2/1 unit 0 description to-R3

```
user@R2# set fe-1/2/1 unit 0 family inet address 10.0.0.5/30
```

```
user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure OSPF or another interior gateway protocol (IGP).

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface fe-1/2/1.0
user@R2# set interface lo0.0 passive
```

3. Configure the EBGP connection to Device R1.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set peer-as 100
user@R2# set neighbor 10.0.0.1
```

4. Configure the IBGP connection to Device R3.

```
[edit protocols bgp group int]
user@R2# set type internal
user@R2# set local-address 192.168.0.2
user@R2# set neighbor 192.168.0.3
```

5. Add the **advertise-external** statement to the IBGP group peering session.

```
[edit protocols bgp group int]
user@R2# set advertise-external
```

6. Configure the autonomous system (AS) number and the router ID.

```
[edit routing-options ]
user@R2# set router-id 192.168.0.2
user@R2# set autonomous-system 200
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    description to-R1;
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    description to-R3;
    family inet {
      address 10.0.0.5/30;
    }
  }
}
lo0 {
```

```

    unit 0 {
      family inet {
        address 192.168.0.2/32;
      }
    }
  }

user@R2# show protocols
bgp {
  group ext {
    type external;
    peer-as 100;
    neighbor 10.0.0.1;
  }
  group int {
    type internal;
    local-address 192.168.0.2;
    advertise-external;
    neighbor 192.168.0.3;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/1.0;
    interface lo0.0 {
      passive;
    }
  }
}

user@R2# show routing-options
router-id 192.168.0.2;
autonomous-system 200;

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying the BGP Active Path on page 309](#)
- [Verifying the External Route Advertisement on page 310](#)
- [Verifying the Route on Device R3 on page 310](#)
- [Experimenting with the conditional Option on page 311](#)

### Verifying the BGP Active Path

**Purpose** On Device R2, make sure that the 172.16.6.0/24 prefix is in the routing table and has the expected active path.

**Action** user@R2> show route 172.16.6

```
inet.0: 8 destinations, 9 routes (8 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.6.0/24      *[BGP/170] 00:00:07, localpref 200, from 192.168.0.3
                   AS path: I, validation-state: unverified
                   > to 10.0.0.6 via fe-1/2/1.0
                   [BGP/170] 03:23:03, localpref 100
                   AS path: 100 I, validation-state: unverified
                   > to 10.0.0.1 via fe-1/2/0.0
```

**Meaning** Device R2 receives the 172.16.6.0/24 route from both Device R1 and Device R3. The route from Device R3 is the active path, as designated by the asterisk (\*). The active path has the highest local preference. Even if the local preferences of the two routes were equal, the route from Device R3 would remain active because it has the shortest AS path.

### Verifying the External Route Advertisement

**Purpose** On Device R2, make sure that the 172.16.6.0/24 route is advertised toward Device R3.

**Action** user@R2> show route advertising-protocol bgp 192.168.0.3

```
inet.0: 8 destinations, 9 routes (8 active, 1 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lclpref   AS path
  172.16.6.0/24         10.0.0.1          100       100       100 I
```

**Meaning** Device R2 is advertising the 172.16.6.0/24 route toward Device R3.

### Verifying the Route on Device R3

**Purpose** Make sure that the 172.16.6.0/24 prefix is in Device R3's routing table.

**Action** user@R3> show route 172.16.6.0/24

```
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.6.0/24      *[Static/5] 03:34:14
                   Reject
                   [BGP/170] 06:34:43, localpref 100, from 192.168.0.2
                   AS path: 100 I, validation-state: unverified
                   > to 10.0.0.5 via fe-1/2/0.6
```

**Meaning** Device R3 has the static route and the BGP route for 172.16.6.0/24.

Note that the BGP route is hidden on Device R3 if the route is not reachable or if the next hop cannot be resolved. To fulfill this requirement, this example includes a static default route on Device R3 (**static route 0.0.0.0/0 next-hop 10.0.0.5**).

### Experimenting with the conditional Option

**Purpose** See how the **conditional** option works in the context of the BGP path selection algorithm.

**Action** 1. On Device R2, add the **conditional** option.

```
[edit protocols bgp group int]
user@R2# set advertise-external conditional
user@R2# commit
```

2. On Device R2, check to see if the 172.16.6.0/24 route is advertised toward Device R3.

```
user@R2> show route advertising-protocol bgp 192.168.0.3
```

As expected, the route is no longer advertised. You might need to wait a few seconds to see this result.

3. On Device R3, deactivate the **then local-preference** policy action.

```
[edit policy-options policy-statement send-static term 1]
user@R3# deactivate logical-systems R3 then local-preference
user@R3# commit
```

4. On Device R2, ensure that the local preferences of the two paths are equal.

```
user@R2> show route 172.16.6.0/24
```

```
inet.0: 8 destinations, 9 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
172.16.6.0/24      *[BGP/170] 08:02:59, localpref 100
                   AS path: 100 I, validation-state: unverified
                   > to 10.0.0.1 via fe-1/2/0.0
                   [BGP/170] 00:07:51, localpref 100, from 192.168.0.3
                   AS path: I, validation-state: unverified
                   > to 10.0.0.6 via fe-1/2/1.0
```

5. On Device R2, add the **as-path-ignore** statement.

```
[edit protocols bgp]
user@R2# set path-selection as-path-ignore
user@R2# commit
```

6. On Device R2, check to see if the 172.16.6.0/24 route is advertised toward Device R3.

```
user@R2> show route advertising-protocol bgp 192.168.0.3
```

```
inet.0: 8 destinations, 9 routes (8 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref    AS path
* 172.16.6.0/24         10.0.0.1          100       100        100 I
```

As expected, the route is now advertised because the AS path length is ignored and because the local preferences are equal.

- Related Documentation**
- [Example: Setting BGP to Advertise Inactive Routes on page 298](#)
  - [Understanding BGP Path Selection on page 8](#)

## Example: Disabling Suppression of Route Advertisements

---

Junos OS does not advertise the routes learned from one EBGp peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGp peers that are in the same autonomous system (AS) as the originating peer, regardless of the routing instance. You can modify this behavior by including the `advertise-peer-as` statement in the configuration.

If you include the `advertise-peer-as` statement in the configuration, BGP advertises the route regardless of this check.

To restore the default behavior, include the `no-advertise-peer-as` statement in the configuration:

```
no-advertise-peer-as;
```

The route suppression default behavior is disabled if the `as-override` statement is included in the configuration. If you include both the `as-override` and `no-advertise-peer-as` statements in the configuration, the `no-advertise-peer-as` statement is ignored.

- [Requirements on page 312](#)
- [Overview on page 312](#)
- [Configuration on page 313](#)
- [Verification on page 317](#)

### Requirements

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

### Overview

This example shows three routing devices with external BGP (EBGP) connections. Device R2 has an EBGp connection to Device R1 and another EBGp connection to Device R3. Although separated by Device R2 which is in AS 64511, Device R1 and Device R3 are in the same AS (AS 64512). Device R1 and Device R3 advertise into BGP direct routes to their own loopback interface addresses.

Device R2 receives these loopback interface routes, and the `advertise peer-as` statement allows Device R2 to advertise them. Specifically, Device R1 sends the 192.168.0.1 route to Device R2, and because Device R2 has the `advertise peer-as` configured, Device R2 can send the 192.168.0.1 route to Device R3. Likewise, Device R3 sends the 192.168.0.3 route to Device R2, and `advertise peer-as` enables Device R2 to forward the route to Device R1.

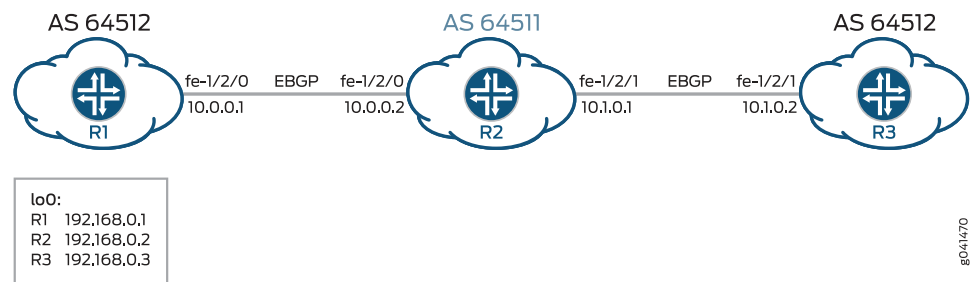
To enable Device R1 and Device R3 to accept routes that contain their own AS number in the AS path, the `loops 2` statement is required on Device R1 and Device R3.

### Topology

---

Figure 25 on page 202 shows the sample network.

Figure 37: BGP Topology for advertise-peer-as



“CLI Quick Configuration” on page 202 shows the configuration for all of the devices in Figure 25 on page 202.

The section “Step-by-Step Procedure” on page 203 describes the steps on Device R1 and 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 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp family inet unicast loops 2
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 300

Device R2
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group ext type external
set protocols bgp group ext advertise-peer-as
set protocols bgp group ext export send-direct
set protocols bgp group ext neighbor 10.0.0.1 peer-as 300
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 200

Device R3
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp family inet unicast loops 2
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.1.0.1

```

```
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 300
```

**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 Device R1:

1. Configure the device interfaces.

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

user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure BGP.

```
[edit protocols bgp group ext]
user@R1# set type external
user@R1# set peer-as 200
user@R1# set neighbor 10.0.0.2
```

3. Prevent routes from Device R3 from being hidden on Device R1 by including the **loops 2** statement.

The **loops 2** statement means that the local device's own AS number can appear in the AS path up to one time without causing the route to be hidden. The route is hidden if the local device's AS number is detected in the path two or more times.

```
[edit protocols bgp family inet unicast]
user@R1# set loops 2
```

4. Configure the routing policy that sends direct routes.

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

5. Apply the export policy to the BGP peering session with Device R2.

```
[edit protocols bgp group ext]
user@R1# set export send-direct
```

6. Configure the autonomous system (AS) number.

```
[edit routing-options ]
user@R1# set autonomous-system 300
```

**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 Device R2:

1. Configure the device interfaces.

```
[edit interfaces]
```

```
user@R2# set fe-1/2/0 unit 0 family inet address 10.0.0.2/30
```

```
user@R2# set fe-1/2/1 unit 0 family inet address 10.1.0.1/30
```

```
user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure BGP.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set neighbor 10.0.0.1 peer-as 300
user@R2# set neighbor 10.1.0.2 peer-as 300
```

3. Configure Device R2 to advertise routes learned from one EBGP peer to another EBGP peer in the same AS.

In other words, advertise to Device R1 routes learned from Device R3 (and the reverse), even though Device R1 and Device R3 are in the same AS.

```
[edit protocols bgp group ext]
user@R2# set advertise-peer-as
```

4. Configure a routing policy that sends direct routes.

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

5. Apply the export policy.

```
[edit protocols bgp group ext]
user@R2# set export send-direct
```

6. Configure the AS number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

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

```
Device R1 user@R1# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
```

```
user@R1# show protocols
bgp {
  family inet {
    unicast {
      loops 2;
    }
  }
  group ext {
    type external;
    export send-direct;
    peer-as 200;
    neighbor 10.0.0.2;
  }
}

user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R1# show routing-options
autonomous-system 300;

Device R2 user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    family inet {
      address 10.1.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.2/32;
    }
  }
}

user@R2# show protocols
bgp {
  group ext {
    type external;
    advertise-peer-as;
    export send-direct;
    neighbor 10.0.0.1 {
      peer-as 300;
    }
  }
}
```

```

    }
    neighbor 10.1.0.2 {
        peer-as 300;
    }
}

user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R2# show routing-options
autonomous-system 200;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the BGP Routes

**Purpose** Make sure that the routing tables on Device R1 and Device R3 contain the expected routes.

**Action** 1. On Device R2, deactivate the **advertise-peer-as** statement in the BGP configuration.

```

[edit protocols bgp group ext]
user@R2# deactivate advertise-peer-as
user@R2# commit

```

2. On Device R3, deactivate the **loops** statement in the BGP configuration.

```

[edit protocols bgp family inet unicast ]
user@R3# deactivate unicast loops
user@R3# commit

```

3. On Device R1, check to see what routes are advertised to Device R2.

```

user@R1> show route advertising-protocol bgp 10.0.0.2
inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
  Prefix                Nexthop                  MED      Lclpref    AS path
* 10.0.0.0/30           Self                      0         0          I
* 192.168.0.1/32       Self                      0         0          I

```

4. On Device R2, check to see what routes are received from Device R1.

```

user@R2> show route receive-protocol bgp 10.0.0.1
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix                Nexthop                  MED      Lclpref    AS path
10.0.0.0/30           10.0.0.1                  0         0          300 I
* 192.168.0.1/32       10.0.0.1                  0         0          300 I

```

5. On Device R2, check to see what routes are advertised to Device R3.

```

user@R2> show route advertising-protocol bgp 10.1.0.2
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix                Nexthop                  MED      Lclpref    AS path

```

```
* 10.0.0.0/30          Self          I
* 10.1.0.0/30          Self          I
* 192.168.0.2/32       Self          I
```

6. On Device R2, activate the **advertise-peer-as** statement in the BGP configuration.

```
[edit protocols bgp group ext]
user@R2# activate advertise-peer-as
user@R2# commit
```

7. On Device R2, recheck the routes that are advertised to Device R3.

```
user@R2> show route advertising-protocol bgp 10.1.0.2
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref  AS path
* 10.0.0.0/30       Self              I
* 10.1.0.0/30       Self              I
* 192.168.0.1/32    Self              300 I
* 192.168.0.2/32    Self              I
* 192.168.0.3/32    10.1.0.2         300 I
```

8. On Device R3, check the routes that are received from Device R2.

```
user@R3> show route receive-protocol bgp 10.1.0.1
inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref  AS path
* 10.0.0.0/30       10.1.0.1         200 I
  10.1.0.0/30       10.1.0.1         200 I
* 192.168.0.2/32    10.1.0.1         200 I
```

9. On Device R3, activate the **loops** statement in the BGP configuration.

```
[edit protocols bgp family inet unicast ]
user@R3# activate unicast loops
user@R3# commit
```

10. On Device R3, recheck the routes that are received from Device R2.

```
user@R3> show route receive-protocol bgp 10.1.0.1
inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 1 hidden)
  Prefix            Nexthop          MED      Lc1pref  AS path
* 10.0.0.0/30       10.1.0.1         200 I
  10.1.0.0/30       10.1.0.1         200 I
* 192.168.0.1/32    10.1.0.1         200 300 I
* 192.168.0.2/32    10.1.0.1         200 I
```

**Meaning** First the **advertise-peer-as** statement and the **loops** statement are deactivated so that the default behavior can be examined. Device R1 sends to Device R2 a route to Device R1's loopback interface address, 192.168.0.1/32. Device R2 does not advertise this route to Device R3. After activating the **advertise-peer-as** statement, Device R2 does advertise the 192.168.0.1/32 route to Device R3. Device R3 does not accept this route until after the **loops** statement is activated.

**Related Documentation**

- [Example: Configuring a Layer 3 VPN with Route Reflection and AS Override on page 191](#)

## Example: Defining a Routing Policy That Removes BGP Communities

This example shows how to create a policy that accepts BGP routes, but removes BGP communities from the routes.

- [Requirements on page 319](#)
- [Overview on page 319](#)
- [Configuration on page 320](#)
- [Verification on page 324](#)

### Requirements

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

### Overview

This example shows two routing devices with an external BGP (EBGP) connection between them. Device R2 uses the BGP session to send two static routes to Device R1. On Device R1, an import policy specifies that all BGP communities must be removed from the routes.

By default, when communities are configured on EBGP peers, they are sent and accepted. To suppress the acceptance of communities received from a neighbor, you can remove all communities or a specified set of communities. When the result of a policy is an empty set of communities, the community attribute is not included. To remove all communities, first define a wildcard set of communities (here, the community is named **wild**):

```
[edit policy-options]
community wild members "*" : *;
```

Then, in the routing policy statement, specify the **community delete** action:

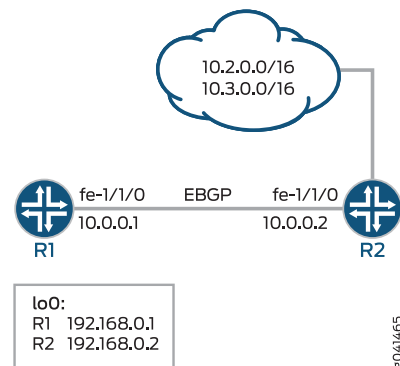
```
[edit policy-options]
policy-statement policy-name {
  term term-name {
    then community delete wild;
  }
}
```

To suppress a particular community from any autonomous system (AS), define the community as **community wild members "\*" : *community-value***.

### Topology

[Figure 38 on page 320](#) shows the sample network.

Figure 38: BGP Policy That Removes Communities



## 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-1/1/0 unit 0 description to-R2
set interfaces fe-1/1/0 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 2
set protocols bgp group external-peers neighbor 10.0.0.2 import remove-communities
set policy-options policy-statement remove-communities term 1 from protocol bgp
set policy-options policy-statement remove-communities term 1 then community delete
  wild
set policy-options policy-statement remove-communities term 1 then accept
set policy-options policy-statement remove-communities term 2 then reject
set policy-options community wild members *:*
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 1

```

**Device R2**

```

set interfaces fe-1/1/0 unit 0 description to-R1
set interfaces fe-1/1/0 unit 0 family inet address 10.0.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers export statics
set protocols bgp group external-peers peer-as 1
set protocols bgp group external-peers neighbor 10.0.0.1
set policy-options policy-statement statics from protocol static
set policy-options policy-statement statics then community add 1
set policy-options policy-statement statics then accept
set policy-options community 1 members 2:1
set policy-options community 1 members 2:2
set policy-options community 1 members 2:3
set policy-options community 1 members 2:4
set policy-options community 1 members 2:5
set policy-options community 1 members 2:6
set policy-options community 1 members 2:7
set policy-options community 1 members 2:8
set policy-options community 1 members 2:9

```

```

set policy-options community 1 members 2:10
set routing-options static route 10.2.0.0/16 reject
set routing-options static route 10.2.0.0/16 install
set routing-options static route 10.3.0.0/16 reject
set routing-options static route 10.3.0.0/16 install
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 2

```

**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 Device R1:

1. Configure the interfaces.

```

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

user@R1# set lo0 unit 0 family inet address 192.168.0.1/32

```

2. Configure BGP.

Apply the import policy to the BGP peering session with Device R2.

```

[edit protocols bgp group external-peers]
user@R1# set type external
user@R1# set peer-as 2
user@R1# set neighbor 10.0.0.2 import remove-communities

```

3. Configure the routing policy that deletes communities.

```

[edit policy-options policy-statement remove-communities]
user@R1# set term 1 from protocol bgp
user@R1# set term 1 then community delete wild
user@R1# set term 1 then accept
user@R1# set term 2 then reject

```

4. Configure the autonomous system (AS) number and the router ID.

```

[edit routing-options ]
user@R1# set router-id 192.168.0.1
user@R1# set autonomous-system 1

```

**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 Device R2:

1. Configure the interfaces.

```

[edit interfaces]
user@R2# set fe-1/1/0 unit 0 description to-R1
user@R2# set fe-1/1/0 unit 0 family inet address 10.0.0.2/30

```

```
user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@R2# set router-id 192.168.0.3
user@R2# set autonomous-system 2
```

3. Configure BGP.

```
[edit protocols bgp group external-peers]
user@R2# set type external
user@R2# set peer-as 1
user@R2# set neighbor 10.0.0.1
```

4. Configure multiple communities, or configure a single community with multiple members.

```
[edit policy-options community 1]
user@R2# set members 2:1
user@R2# set members 2:2
user@R2# set members 2:3
user@R2# set members 2:4
user@R2# set members 2:5
user@R2# set members 2:6
user@R2# set members 2:7
user@R2# set members 2:8
user@R2# set members 2:9
user@R2# set members 2:10
```

5. Configure the static routes.

```
[edit routing-options static]
user@R2# set route 10.2.0.0/16 reject
user@R2# set route 10.2.0.0/16 install
user@R2# set route 10.3.0.0/16 reject
user@R2# set route 10.3.0.0/16 install
```

6. Configure a routing policy that advertises static routes into BGP and adds the BGP community to the routes.

```
[edit policy-options policy-statement statics]
user@R2# set from protocol static
user@R2# set then community add 1
user@R2# set then accept
```

7. Apply the export policy.

```
[edit protocols bgp group external-peers]
user@R2# set export statics
```

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

**Device R1**

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

```

        description to-R2;
        family inet {
            address 10.0.0.1/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.1/32;
        }
    }
}

user@R1# show protocols
bgp {
    group external-peers {
        type external;
        peer-as 2;
        neighbor 10.0.0.2 {
            import remove-communities;
        }
    }
}

user@R1# show policy-options
policy-statement remove-communities {
    term 1 {
        from protocol bgp;
        then {
            community delete wild;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
community wild members *.*;

user@R1# show routing-options
router-id 192.168.0.1;
autonomous-system 1;

Device R2 user@R2# show interfaces
fe-1/1/0 {
    unit 0 {
        description to-R1;
        family inet {
            address 10.0.0.2/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.2/32;
        }
    }
}

```

```

    }
  }
}

user@R2# show protocols
bgp {
  group external-peers {
    type external;
    export statics;
    peer-as 1;
    neighbor 10.0.0.1;
  }
}

user@R2# show policy-options
policy-statement statics {
  from protocol static;
  then {
    community add 1;
    accept;
  }
}
community 1 members [ 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10 ];

user@R2# show routing-options
static {
  route 10.2.0.0/16 {
    reject;
    install;
  }
  route 10.3.0.0/16 {
    reject;
    install;
  }
}
router-id 192.168.0.3;
autonomous-system 2;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the BGP Routes

**Purpose** Make sure that the routing table on Device R1 does not contain BGP communities.

**Action** 1. On Device R1, run the **show route protocols bgp extensive** command.

```

user@R1> show route protocols bgp extensive

inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
10.2.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.2.0.0/16 -> {10.0.0.2}
*BGP Preference: 170/-101

```

```

Next hop type: Router, Next hop index: 671
Address: 0x9458270
Next-hop reference count: 4
Source: 10.0.0.2
Next hop: 10.0.0.2 via lt-1/1/0.5, selected
Session Id: 0x100001
State: <Active Ext>
Local AS: 1 Peer AS: 2
Age: 20:39:01
Validation State: unverified
Task: BGP_2.10.0.0.2+179
Announcement bits (1): 0-KRT
AS path: 2 I
Accepted
Localpref: 100
Router ID: 192.168.0.3

```

10.3.0.0/16 (1 entry, 1 announced)

TSI:

KRT in-kerne1 10.3.0.0/16 -> {10.0.0.2}

```

*BGP Preference: 170/-101
Next hop type: Router, Next hop index: 671
Address: 0x9458270
Next-hop reference count: 4
Source: 10.0.0.2
Next hop: 10.0.0.2 via lt-1/1/0.5, selected
Session Id: 0x100001
State: <Active Ext>
Local AS: 1 Peer AS: 2
Age: 20:39:01
Validation State: unverified
Task: BGP_2.10.0.0.2+179
Announcement bits (1): 0-KRT
AS path: 2 I
Accepted
Localpref: 100
Router ID: 192.168.0.3

```

2. On Device R1, deactivate the **community remove** configuration in the import policy.

```

[edit policy-options policy-statement remove-communities term 1]
user@R1# deactivate then community delete wild
user@R1# commit

```

3. On Device R1, run the **show route protocols bgp extensive** command to view the advertised communities.

```

user@R1> show route protocols bgp extensive
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
10.2.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kerne1 10.2.0.0/16 -> {10.0.0.2}
*BGP Preference: 170/-101
Next hop type: Router, Next hop index: 671
Address: 0x9458270
Next-hop reference count: 4
Source: 10.0.0.2
Next hop: 10.0.0.2 via lt-1/1/0.5, selected
Session Id: 0x100001
State: <Active Ext>
Local AS: 1 Peer AS: 2
Age: 20:40:53

```

```

Validation State: unverified
Task: BGP_2.10.0.0.2+179
Announcement bits (1): 0-KRT
AS path: 2 I
Communities: 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10
Accepted
Localpref: 100
Router ID: 192.168.0.3

10.3.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.3.0.0/16 -> {10.0.0.2}
  *BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 671
        Address: 0x9458270
        Next-hop reference count: 4
        Source: 10.0.0.2
        Next hop: 10.0.0.2 via lt-1/1/0.5, selected
        Session Id: 0x100001
        State: <Active Ext>
        Local AS:      1 Peer AS:      2
        Age: 20:40:53
        Validation State: unverified
        Task: BGP_2.10.0.0.2+179
        Announcement bits (1): 0-KRT
        AS path: 2 I
        Communities: 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10
        Accepted
        Localpref: 100
        Router ID: 192.168.0.3

```

**Meaning** The output shows that in Device R1's routing table, the communities are suppressed in the BGP routes sent from Device R2. When the **community remove** setting in Device R1's import policy is deactivated, the communities are no longer suppressed.

**Related Documentation**

- [Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS on page 225](#)
- [Understanding External BGP Peering Sessions on page 33](#)

## Example: Defining a Routing Policy Based on the Number of BGP Communities

This example shows how to create a policy that accepts BGP routes based on the number of BGP communities.

- [Requirements on page 327](#)
- [Overview on page 327](#)
- [Configuration on page 327](#)
- [Verification on page 332](#)

## Requirements

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

## Overview

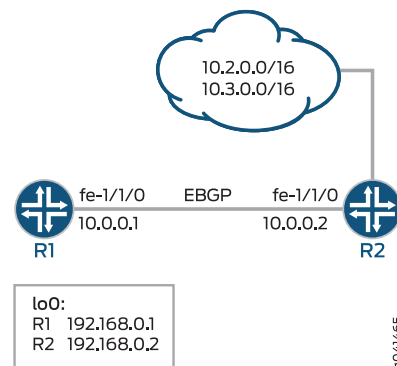
This example shows two routing devices with an external BGP (EBGP) connection between them. Device R2 uses the BGP session to send two static routes to Device R1. On Device R1, an import policy specifies that the BGP-received routes can contain up to five communities to be considered a match. For example, if a route contains three communities, it is considered a match and is accepted. If a route contains six or more communities, it is considered a nonmatch and is rejected.

It is important to remember that the default policy for EBGP is to accept all routes. To ensure that the nonmatching routes are rejected, you must include a **then reject** action at the end of the policy definition.

## Topology

Figure 39 on page 327 shows the sample network.

**Figure 39: BGP Policy with a Limit on the Number of Communities Accepted**



## 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-1/1/0 unit 0 description to-R2
set interfaces fe-1/1/0 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 2
set protocols bgp group external-peers neighbor 10.0.0.2 import import-communities
set policy-options policy-statement import-communities term 1 from protocol bgp
set policy-options policy-statement import-communities term 1 from community-count
5 orlower
```

```
set policy-options policy-statement import-communities term 1 then accept
set policy-options policy-statement import-communities term 2 then reject
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 1
```

**Device R2**

```
set interfaces fe-1/1/0 unit 0 description to-R1
set interfaces fe-1/1/0 unit 0 family inet address 10.0.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers export statics
set protocols bgp group external-peers peer-as 1
set protocols bgp group external-peers neighbor 10.0.0.1
set policy-options policy-statement statics from protocol static
set policy-options policy-statement statics then community add 1
set policy-options policy-statement statics then accept
set policy-options community 1 members 2:1
set policy-options community 1 members 2:2
set policy-options community 1 members 2:3
set policy-options community 1 members 2:4
set policy-options community 1 members 2:5
set policy-options community 1 members 2:6
set policy-options community 1 members 2:7
set policy-options community 1 members 2:8
set policy-options community 1 members 2:9
set policy-options community 1 members 2:10
set routing-options static route 10.2.0.0/16 reject
set routing-options static route 10.2.0.0/16 install
set routing-options static route 10.3.0.0/16 reject
set routing-options static route 10.3.0.0/16 install
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 2
```

**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 Device R1:

1. Configure the interfaces.

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

user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure BGP.

Apply the import policy to the BGP peering session with Device R2.

```
[edit protocols bgp group external-peers]
user@R1# set type external
user@R1# set peer-as 2
user@R1# set neighbor 10.0.0.2 import import-communities
```

3. Configure the routing policy that sends direct routes.

```
[edit policy-options policy-statement import-communities]
user@R1# set term 1 from protocol bgp
user@R1# set term 1 from community-count 5 orlower
user@R1# set term 1 then accept
user@R1# set term 2 then reject
```

4. Configure the autonomous system (AS) number and the router ID.

```
[edit routing-options ]
user@R1# set router-id 192.168.0.1
user@R1# set autonomous-system 1
```

### 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 Device R2:

1. Configure the interfaces.

```
[edit interfaces]
user@R2# set fe-1/1/0 unit 0 description to-R1
user@R2# set fe-1/1/0 unit 0 family inet address 10.0.0.2/30

user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@R2# set router-id 192.168.0.3
user@R2# set autonomous-system 2
```

3. Configure BGP.

```
[edit protocols bgp group external-peers]
user@R2# set type external
user@R2# set peer-as 1
user@R2# set neighbor 10.0.0.1
```

4. Configure multiple communities, or configure a single community with multiple members.

```
[edit policy-options community 1]
user@R2# set members 2:1
user@R2# set members 2:2
user@R2# set members 2:3
user@R2# set members 2:4
user@R2# set members 2:5
user@R2# set members 2:6
user@R2# set members 2:7
user@R2# set members 2:8
user@R2# set members 2:9
user@R2# set members 2:10
```

5. Configure the static routes.

```
[edit routing-options static]
user@R2# set route 10.2.0.0/16 reject
user@R2# set route 10.2.0.0/16 install
```

```

user@R2# set route 10.3.0.0/16 reject
user@R2# set route 10.3.0.0/16 install

```

6. Configure a routing policy that advertises static routes into BGP and adds the BGP community to the routes.

```

[edit policy-options policy-statement statics]
user@R2# set from protocol static
user@R2# set then community add 1
user@R2# set then accept

```

7. Apply the export policy.

```

[edit protocols bgp group external-peers]
user@R2# set export statics

```

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

```

Device R1 user@R1# show interfaces
fe-1/1/0 {
  unit 0 {
    description to-R2;
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
}

user@R1# show protocols
bgp {
  group external-peers {
    type external;
    peer-as 2;
    neighbor 10.0.0.2 {
      import import-communities;
    }
  }
}

user@R1# show policy-options
policy-statement import-communities {
  term 1 {
    from {
      protocol bgp;
      community-count 5 orlower;
    }
  }
}

```

```

        then accept;
    }
    term 2 {
        then reject;
    }
}

user@R1# show routing-options
router-id 192.168.0.1;
autonomous-system 1;

Device R2 user@R2# show interfaces
fe-1/1/0 {
    unit 0 {
        description to-R1;
        family inet {
            address 10.0.0.2/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.2/32;
        }
    }
}

user@R2# show protocols
bgp {
    group external-peers {
        type external;
        export statics;
        peer-as 1;
        neighbor 10.0.0.1;
    }
}

user@R2# show policy-options
policy-statement statics {
    from protocol static;
    then {
        community add 1;
        accept;
    }
}
community 1 members [ 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10 ];

user@R2# show routing-options
static {
    route 10.2.0.0/16 {
        reject;
        install;
    }
    route 10.3.0.0/16 {
        reject;
        install;
    }
}

```

```

    }
  }
  router-id 192.168.0.3;
  autonomous-system 2;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the BGP Routes

**Purpose** Make sure that the routing table on Device R1 contains the expected BGP routes.

**Action** 1. On Device R1, run the **show route protocols bgp** command.

```
user@R1> show route protocols bgp
```

```
inet.0: 5 destinations, 5 routes (3 active, 0 holddown, 2 hidden)
```

2. On Device R1, change the **community-count** configuration in the import policy.

```

[edit policy-options policy-statement import-communities term 1]
user@R1# set from community-count 5 orhigher
user@R1# commit

```

3. On Device R1, run the **show route protocols bgp** command.

```
user@R1> show route protocols bgp
```

```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.2.0.0/16      *[BGP/170] 18:29:53, localpref 100
                  AS path: 2 I, validation-state: unverified
                  > to 10.0.0.2 via fe-1/1/0.0
10.3.0.0/16      *[BGP/170] 18:29:53, localpref 100
                  AS path: 2 I, validation-state: unverified
                  > to 10.0.0.2 via fe-1/1/0.0

```

4. On Device R1, run the **show route protocols bgp extensive** command to view the advertised communities.

```

user@R1> show route protocols bgp extensive
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
10.2.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.2.0.0/16 -> {10.0.0.2}
  *BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 671
        Address: 0x9458270
        Next-hop reference count: 4
        Source: 10.0.0.2
        Next hop: 10.0.0.2 via fe-1/1/0.0, selected
        Session Id: 0x100001
        State: <Active Ext>
        Local AS:      1 Peer AS:      2
        Age: 18:56:10
        Validation State: unverified

```

```

Task: BGP_2.10.0.0.2+179
Announcement bits (1): 0-KRT
AS path: 2 I
Communities: 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10
Accepted
Localpref: 100
Router ID: 192.168.0.3

10.3.0.0/16 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.3.0.0/16 -> {10.0.0.2}
  *BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 671
    Address: 0x9458270
    Next-hop reference count: 4
    Source: 10.0.0.2
    Next hop: 10.0.0.2 via fe-1/1/0.0, selected
    Session Id: 0x100001
    State: <Active Ext>
    Local AS: 1 Peer AS: 2
    Age: 18:56:10
    Validation State: unverified
    Task: BGP_2.10.0.0.2+179
    Announcement bits (1): 0-KRT
    AS path: 2 I
    Communities: 2:1 2:2 2:3 2:4 2:5 2:6 2:7 2:8 2:9 2:10
    Accepted
    Localpref: 100
    Router ID: 192.168.0.3

```

**Meaning** The output shows that in Device R1's routing table, the BGP routes sent from Device R2 are hidden. When the **community-count** setting in Device R1's import policy is modified, the BGP routes are no longer hidden.

**Related Documentation**

- [Example: Redistributing BGP Routes with a Specific Community Tag into IS-IS on page 225](#)
- [Understanding External BGP Peering Sessions on page 33](#)

## Example: Using Routing Policy to Set a Preference Value for BGP Routes

This example shows how to use routing policy to set the preference for routes learned from BGP. Routing information can be learned from multiple sources. To break ties among equally specific routes learned from multiple sources, each source has a preference value. Routes that are learned through explicit administrative action, such as static routes, are preferred over routes learned from a routing protocol, such as BGP or OSPF. This concept is called *administrative distance* by some vendors.

- [Requirements on page 334](#)
- [Overview on page 334](#)
- [Configuration on page 335](#)
- [Verification on page 338](#)

## Requirements

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

## Overview

Routing information can be learned from multiple sources, such as through static configuration, BGP, or an interior gateway protocol (IGP). When Junos OS determines a route's preference to become the active route, it selects the route with the lowest preference as the active route and installs this route into the forwarding table. By default, the routing software assigns a preference of 170 to routes that originated from BGP. Of all the routing protocols, BGP has the highest default preference value, which means that routes learned by BGP are the least likely to become the active route.

Some vendors have a preference (distance) of 20 for external BGP (EBGP) and a distance of 200 for internal BGP (IBGP). Junos OS uses the same value (170) for both EBGP and IBGP. However, this difference between vendors has no operational impact because Junos OS always prefers EBGP routes over IBGP routes.

Another area in which vendors differ is in regard to IGP distance compared to BGP distance. For example, some vendors assign a distance of 110 to OSPF routes. This is higher than the EBGP distance of 20, and results in the selection of an EBGP route over an equivalent OSPF route. In the same scenario, Junos OS chooses the OSPF route, because of the default preference 10 for an internal OSPF route and 150 for an external OSPF route, which are both lower than the 170 preference assigned to all BGP routes.

This example shows a routing policy that matches routes from specific next hops and sets a preference. If a route does not match the first term, it is evaluated by the second term.

---

### Topology

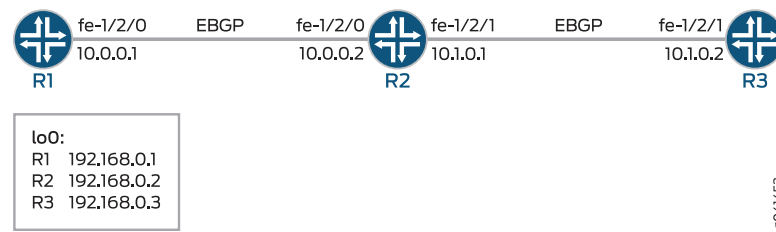
In the sample network, Device R1 and Device R3 have EBGP sessions with Device R2.

On Device R2, an import policy takes the following actions:

- For routes received through BGP from next-hop 10.0.0.1 (Device R1), set the route preference to 10.
- For routes received through BGP from next-hop 10.1.0.2 (Device R3), set the route preference to 15.

[Figure 40 on page 335](#) shows the sample network.

Figure 40: BGP Preference Value Topology



“CLI Quick Configuration” on page 335 shows the configuration for all of the devices in Figure 40 on page 335.

The section “Step-by-Step Procedure” on page 336 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 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 100
```

**Device R2**

```
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group ext type external
set protocols bgp group ext import set-preference
set protocols bgp group ext export send-direct
set protocols bgp group ext neighbor 10.0.0.1 peer-as 100
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement set-preference term term1 from protocol bgp
set policy-options policy-statement set-preference term term1 from next-hop 10.0.0.1
set policy-options policy-statement set-preference term term1 then preference 10
set policy-options policy-statement set-preference term term2 from protocol bgp
set policy-options policy-statement set-preference term term2 from next-hop 10.1.0.2
set policy-options policy-statement set-preference term term2 then preference 15
set routing-options autonomous-system 200
```

**Device R3**

```
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
```

```
set protocols bgp group ext neighbor 10.1.0.1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 300
```

**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 Device R2:

1. Configure the device interfaces.

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

user@R2# set fe-1/2/1 unit 0 family inet address 10.1.0.1/30

user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure the local autonomous system.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

3. Configure the routing policy that sends direct routes.

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

4. Configure the routing policy that changes the preference of received routes.

```
[edit policy-options policy-statement set-preference]
user@R2# set term term1 from protocol bgp
user@R2# set term term1 from next-hop 10.0.0.1
user@R2# set term term1 then preference 10
```

```
user@R2# set term term2 from protocol bgp
user@R2# set term term2 from next-hop 10.1.0.2
user@R2# set term term2 then preference 15
```

5. Configure the external peering with Device R2.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set export send-direct
user@R2# set neighbor 10.0.0.1 peer-as 100
user@R2# set neighbor 10.1.0.2 peer-as 300
```

6. Apply the **set-preference** policy as an import policy.

This affects Device R2's routing table and has no impact on Device R1 and Device R3.

```
[edit protocols bgp group ext]
user@R2# set import set-preference
```

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

```

user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    family inet {
      address 10.1.0.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.2/32;
    }
  }
}
}

user@R2# show protocols
bgp {
  group ext {
    type external;
    import set-preference;
    export send-direct;
    neighbor 10.0.0.1 {
      peer-as 100;
    }
    neighbor 10.1.0.2 {
      peer-as 300;
    }
  }
}

user@R2# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement set-preference {
  term term1 {
    from {
      protocol bgp;
      next-hop 10.0.0.1;
    }
  }
}

```

```

    then {
        preference 10;
    }
}
term term2 {
    from {
        protocol bgp;
        next-hop 10.1.0.2;
    }
    then {
        preference 15;
    }
}
}

```

```

user@R2# show routing-options
autonomous-system 200;

```

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

## Verification

Confirm that the configuration is working properly.

### Verifying the Preference

**Purpose** Make sure that the routing tables on Device R1 and Device R2 reflect the fact that Device R1 is using the configured EBGp preference of 8, and Device R2 is using the default EBGp preference of 170.

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

```

user@R2> show route protocols bgp
inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30          [BGP/10] 04:42:23, localpref 100
                    AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
10.1.0.0/30          [BGP/15] 04:42:23, localpref 100
                    AS path: 300 I, validation-state: unverified
                    > to 10.1.0.2 via fe-1/2/1.0
192.168.0.1/32       *[BGP/10] 04:42:23, localpref 100
                    AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
192.168.0.3/32       *[BGP/15] 04:42:23, localpref 100
                    AS path: 300 I, validation-state: unverified
                    > to 10.1.0.2 via fe-1/2/1.0

```

**Meaning** The output shows that on Device R2, the preference values have been changed to 15 for routes learned from Device R3, and the preference values have been changed to 10 for routes learned from Device R1.

**Related Documentation**

- [Route Preferences Overview](#)
- [Understanding External BGP Peering Sessions on page 33](#)

## Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths

This example shows how to configure BGP to select multiple unequal-cost paths as active paths.

BGP communities can help you control routing policy. An example of a good use for BGP communities is unequal load balancing. When an autonomous system border router (ASBR) receives routes from directly connected external BGP (EBGP) neighbors, the ASBR then advertises those routes to internal neighbors, using IBGP advertisements. In the IBGP advertisements, you can attach the link-bandwidth community to communicate the bandwidth of the advertised external link. This is useful when multiple external links are available, and you want to do unequal load balancing over the links. You configure the link-bandwidth extended community on all ingress links of the AS. The bandwidth information in the link-bandwidth extended community is based on the configured bandwidth of the EBGP link. It is not based on the amount of traffic on the link. Junos OS supports BGP link-bandwidth and multipath load balancing, as described in Internet draft draft-ietf-idr-link-bandwidth-06, *BGP Link Bandwidth Extended Community*.

- [Requirements on page 339](#)
- [Overview on page 339](#)
- [Configuration on page 341](#)
- [Verification on page 345](#)

### Requirements

Before you begin:

- Configure the device interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure a routing policy that exports routes (such as direct routes or IGP routes) from the routing table into BGP.

### Overview

In this example, Device R1 is in AS 65000 and is connected to both Device R2 and Device R3, which are in AS 65001.

The example uses the bandwidth extended community.

By default, when BGP multipath is used, traffic is distributed equally among the several paths calculated. The bandwidth extended community allows an additional attribute to be added to BGP paths, thus allowing the traffic to be distributed unequally. The primary application is a scenario where multiple external paths exist for a given network with asymmetric bandwidth capabilities. In such a scenario, you can tag routes received with the bandwidth extended community. When BGP multipath (internal or external) operates among routes that contain the bandwidth attribute, the forwarding engine can unequally distribute traffic according to the bandwidth corresponding to each path.

When BGP has several candidate paths available for multipath purposes, BGP does not perform unequal cost load balancing according to the bandwidth community unless all candidate paths have this attribute.

The applicability of the bandwidth extended community is limited by the restrictions under which BGP multipath accepts multiple paths for consideration. Explicitly, the IGP distance, as far as BGP is concerned, between the router performing load balancing and the multiple exit points needs to be the same. This can be achieved by using a full mesh of label-switched paths (LSPs) that do not track the corresponding IGP metric. However, in a network in which the propagation delay of circuits is significant (for example, if long-haul circuits are present), it is often valuable to take into account the delay characteristics of different paths.

Configure the bandwidth community as follows:

```
[edit policy-options]
user@host# set community members bandwidth:[1-65535]:[0-4294967295]
```

The first 16-bit number represents the local autonomous system. The second 32-bit number represents the link bandwidth in bytes per second.

For example:

```
[edit policy-options]
user@host# show
community bw-t1 members bandwidth:10458:193000;
community bw-t3 members bandwidth:10458:5592000;
community bw-oc3 members bandwidth:10458:19440000;
```

Where 10458 is the local AS number. The values correspond to the bandwidth of the T1, T3, and OC-3 paths in bytes per second. The value specified as the bandwidth value does not need to correspond to the actual bandwidth of a specific interface. The balance factors used are calculated as a function of the total bandwidth specified. To tag a route with this extended community, define a policy statement, as follows:

```
[edit policy-options]
user@host# show
policy-statement link-bw-t1 {
  then {
    community set bw-t1;
  }
  accept;
}
```

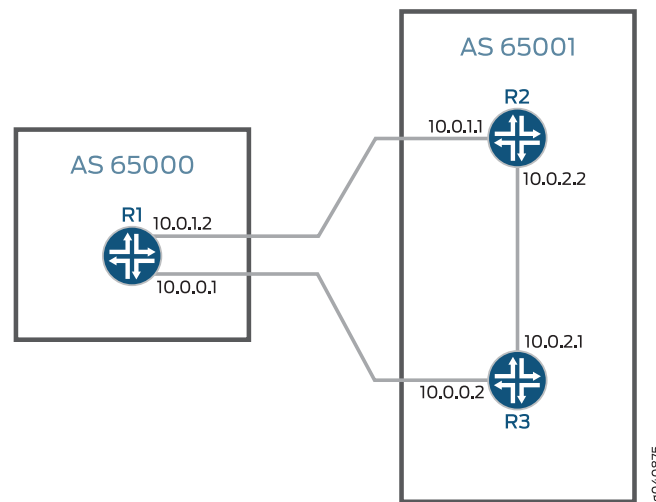
Apply this as an import policy on the BGP peering sessions facing the asymmetrical bandwidth links. Although in theory the community attribute can be added or removed at any point in the network, in the scenario described above, applying the community as an import policy in the EBGP peering session facing the external link allows for that attribute to influence the local multipath decision, and is potentially easier to manage.

---

## Topology

Figure 41 on page 341 shows the topology used in this example.

Figure 41: BGP Load Balancing



“CLI Quick Configuration” on page 341 shows the configuration for all of the devices in Figure 41 on page 341. The section “Step-by-Step Procedure” on page 342 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 ge-1/2/0 unit 0 description R1->R3
set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces ge-1/2/1 unit 0 description R1->R2
set interfaces ge-1/2/1 unit 0 family inet address 10.0.1.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group external type external
set protocols bgp group external import bw-dis
set protocols bgp group external peer-as 65001
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.1
set protocols bgp group external neighbor 10.0.0.2
set policy-options policy-statement bw-dis term a from protocol bgp
set policy-options policy-statement bw-dis term a from neighbor 10.0.1.1
set policy-options policy-statement bw-dis term a then community add bw-high
set policy-options policy-statement bw-dis term a then accept
set policy-options policy-statement bw-dis term b from protocol bgp
set policy-options policy-statement bw-dis term b from neighbor 10.0.0.2
set policy-options policy-statement bw-dis term b then community add bw-low
set policy-options policy-statement bw-dis term b then accept
set policy-options policy-statement loadbal from route-filter 10.0.0.0/16 orlonger
set policy-options policy-statement loadbal then load-balance per-packet
set policy-options community bw-high members bandwidth:65000:60000000
set policy-options community bw-low members bandwidth:65000:40000000
set routing-options autonomous-system 65000
set routing-options forwarding-table export loadbal

```

**Device R2**

```
set interfaces ge-1/2/0 unit 0 description R2->R1
set interfaces ge-1/2/0 unit 0 family inet address 10.0.1.1/30
set interfaces ge-1/2/1 unit 0 description R2->R3
set interfaces ge-1/2/1 unit 0 family inet address 10.0.2.2/30
set interfaces ge-1/2/1 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set interfaces lo0 unit 0 family iso address 49.0001.1921.6800.0002.00
set protocols bgp group external type external
set protocols bgp group external export bgp-default
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 65000
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.2
set protocols isis interface ge-1/2/1.0
set protocols isis interface lo0.0
set policy-options policy-statement bgp-default from protocol static
set policy-options policy-statement bgp-default from route-filter 172.16.0.0/16 exact
set policy-options policy-statement bgp-default then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options static route 172.16.0.0/16 discard
set routing-options static route 172.16.0.0/16 no-install
set routing-options autonomous-system 65001
```

**Device R3**

```
set interfaces ge-1/2/0 unit 0 description R3->R2
set interfaces ge-1/2/0 unit 0 family inet address 10.0.2.1/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces ge-1/2/1 unit 0 description R3->R1
set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set interfaces lo0 unit 0 family iso address 49.0001.1921.6800.0003.00
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external export bgp-default
set protocols bgp group external peer-as 65000
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.0.1
set protocols isis interface ge-1/2/0.0
set protocols isis interface lo0.0
set policy-options policy-statement bgp-default from protocol static
set policy-options policy-statement bgp-default from route-filter 172.16.0.0/16 exact
set policy-options policy-statement bgp-default then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options static route 172.16.0.0/16 discard
set routing-options static route 172.16.0.0/16 no-install
set routing-options autonomous-system 65001
```

**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 BGP peer sessions:

1. Configure the interfaces.

```
user@R1# set ge-1/2/0 unit 0 description R1->R3
user@R1# set ge-1/2/0 unit 0 family inet address 10.0.0.1/30
```

```
user@R1# set ge-1/2/1 unit 0 description R1->R2
user@R1# set ge-1/2/1 unit 0 family inet address 10.0.1.2/30
```

```
user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure the BGP group.

```
[edit protocols bgp group external]
user@R1# set type external
user@R1# set import bw-dis
user@R1# set peer-as 65001
user@R1# set neighbor 10.0.1.1
user@R1# set neighbor 10.0.0.2
```

3. Enable the BGP group to use multiple paths.



**NOTE:** To disable the default check requiring that paths accepted by BGP multipath must have the same neighboring autonomous system (AS), include the `multiple-as` option. Use the `multiple-as` option if the neighbors are in different ASs.

```
[edit protocols bgp group external]
user@R1# set multipath
```

4. Configure the load-balancing policy.

```
[edit policy-options policy-statement loadbal]
user@R1# set from route-filter 10.0.0.0/16 orlonger
user@R1# set then load-balance per-packet
```

5. Apply the load-balancing policy.

```
[edit routing-options]
user@R1# set forwarding-table export loadbal
```

6. Configure the BGP community members.

This example assumes a bandwidth of 1 Gbps and allocates 60 percent to bw-high and 40 percent to bw-low. The reference bandwidth does not need to be the same as the link bandwidth.

```
[editpolicy-options]
user@R1# set community bw-high members bandwidth:65000:600000000
user@R1# set community bw-low members bandwidth:65000:400000000
```

7. Configure the bandwidth distribution policy.

```
[editpolicy-options bw-dis]
user@R1# set term a from protocol bgp
user@R1# set term a from neighbor 10.0.1.1
user@R1# set term a then community add bw-high
user@R1# set term a then accept
```

```

user@R1# set term b from protocol bgp
user@R1# set term b from neighbor 10.0.0.2
user@R1# set term b then community add bw-low
user@R1# set term b then accept

```

8. Configure the local autonomous system (AS) number.

```

[edit routing-options]
user@R1# set autonomous-system 65000

```

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

```

user@R1# show interfaces
ge-1/2/0 {
  unit 0 {
    description R1->R3;
    family inet {
      address 10.0.0.1/30;
    }
  }
}
ge-1/2/1 {
  unit 0 {
    description R1->R2;
    family inet {
      address 10.0.1.2/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}

user@R1# show protocols
bgp {
  group external {
    type external;
    import bw-dis;
    peer-as 65001;
    multipath;
    neighbor 10.0.1.1;
    neighbor 10.0.0.2;
  }
}

user@R1# show policy-options
policy-statement bw-dis {
  term a {
    from {

```

```

        protocol bgp;
        neighbor 10.0.1.1;
    }
    then {
        community add bw-high;
        accept;
    }
}
term b {
    from {
        protocol bgp;
        neighbor 10.0.0.2;
    }
    then {
        community add bw-low;
        accept;
    }
}
}
policy-statement loadbal {
    from {
        route-filter 10.0.0.0/16 orlonger;
    }
    then {
        load-balance per-packet;
    }
}
community bw-high members bandwidth:65000:600000000;
community bw-low members bandwidth:65000:400000000;

user@R1# show routing-options
autonomous-system 65000;
forwarding-table {
    export loadbal;
}

```

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

## Verification

Confirm that the configuration is working properly:

### Verifying Routes

**Purpose** Verify that both routes are selected and that the next hops on the routes show a 60%/40% balance.

**Action** From operational mode, run the **show route protocol bgp detail** command.

```

user@R1> show route 172.16/16 protocol bgp detail
inet.0: 9 destinations, 13 routes (9 active, 0 holddown, 0 hidden)
172.16.0.0/16 (2 entries, 1 announced)
    *BGP      Preference: 170/-101
                Next hop type: Router, Next hop index: 262143
                Address: 0x93fc078
                Next-hop reference count: 3

```

```

Source: 10.0.0.2
Next hop: 10.0.0.2 via ge-1/2/0.0 balance 40%
Next hop: 10.0.1.1 via ge-1/2/1.0 balance 60%, selected
State: **Active Ext>
Local AS: 65000 Peer AS: 65001
Age: 3:22:55
Task: BGP_65001.10.0.0.2+55344
Announcement bits (1): 0-KRT
AS path: 65001 I
Communities: bandwidth:65000:40000000
Accepted Multipath
Localpref: 100
Router ID: 192.168.0.3
BGP Preference: 170/-101
Next hop type: Router, Next hop index: 658
Address: 0x9260520
Next-hop reference count: 4
Source: 10.0.1.1
Next hop: 10.0.1.1 via ge-1/2/1.0, selected
State: <NotBest Ext>
Inactive reason: Not Best in its group - Active preferred
Local AS: 65000 Peer AS: 65001
Age: 3:22:55
Task: BGP_65001.10.0.1.1+62586
AS path: 65001 I
Communities: bandwidth:65000:60000000
Accepted MultipathContrib
Localpref: 100
Router ID: 192.168.0.2

```

```

user@R1> show route 10.0.2.0 protocol bgp detail
inet.0: 9 destinations, 13 routes (9 active, 0 holddown, 0 hidden)
10.0.2.0/30 (2 entries, 1 announced)
*BGP Preference: 170/-101
Next hop type: Router, Next hop index: 262143
Address: 0x93fc078
Next-hop reference count: 3
Source: 10.0.1.1
Next hop: 10.0.0.2 via ge-1/2/0.0 balance 40%
Next hop: 10.0.1.1 via ge-1/2/1.0 balance 60%, selected
State: <Active Ext>
Local AS: 65000 Peer AS: 65001
Age: 3:36:37
Task: BGP_65001.10.0.1.1+62586
Announcement bits (1): 0-KRT
AS path: 65001 I
Communities: bandwidth:65000:60000000
Accepted Multipath
Localpref: 100
Router ID: 192.168.0.2
BGP Preference: 170/-101
Next hop type: Router, Next hop index: 657
Address: 0x92604d8
Next-hop reference count: 4
Source: 10.0.0.2
Next hop: 10.0.0.2 via ge-1/2/0.0, selected
State: <NotBest Ext>
Inactive reason: Not Best in its group - Active preferred
Local AS: 65000 Peer AS: 65001
Age: 3:36:36
Task: BGP_65001.10.0.0.2+55344

```

```
AS path: 65001 I
Communities: bandwidth:65000:40000000
Accepted MultipathContrib
Localpref: 100
Router ID: 192.168.0.3
```

**Meaning** The active path, denoted with an asterisk (\*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 172.16/16 destination.

Likewise, the active path, denoted with an asterisk (\*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 10.0.2.0 destination.

In both cases, the 10.0.1.1 next hop is copied from the inactive path to the active path.

The balance of 40 percent and 60 percent is shown in the **show route** output. This indicates that traffic is being distributed between two next hops and that 60 percent of the traffic is following the first path, while 40 percent is following the second path.

**Related Documentation**

- [Understanding BGP Multipath on page 365](#)



## CHAPTER 7

# BGP BFD Configuration

- [Example: Configuring BFD for BGP on page 349](#)
- [Example: Configuring BFD Authentication for BGP on page 358](#)

### Example: Configuring BFD for BGP

---

- [Understanding BFD for BGP on page 349](#)
- [Example: Configuring BFD on Internal BGP Peer Sessions on page 350](#)

### Understanding BFD for BGP

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. The failure detection timers for BFD have shorter time limits than default failure detection mechanisms for BGP, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

In Junos OS Release 8.3 and later, BFD is supported on internal BGP (IBGP) and multihop external BGP (EBGP) sessions as well as on single-hop EBGP sessions. In Junos OS Release 9.1 through Junos OS Release 11.1, BFD supports IPv6 interfaces in static routes only. In Junos OS Release 11.2 and later, BFD supports IPv6 interfaces with BGP.

## Example: Configuring BFD on Internal BGP Peer Sessions

This example shows how to configure internal BGP (IBGP) peer sessions with the Bidirectional Forwarding Detection (BFD) protocol to detect failures in a network.

- [Requirements on page 350](#)
- [Overview on page 350](#)
- [Configuration on page 351](#)
- [Verification on page 355](#)

---

### Requirements

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

---

### Overview

The minimum configuration to enable BFD on IBGP sessions is to include the **bfd-liveness-detection minimum-interval** statement in the BGP configuration of all neighbors participating in the BFD session. The **minimum-interval** statement specifies the minimum transmit and receive intervals for failure detection. Specifically, this value represents the minimum interval after which the local routing device transmits hello packets as well as the minimum interval that the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value from 1 through 255,000 milliseconds.

Optionally, you can specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and **minimum-receive-interval** statements. For information about these and other optional BFD configuration statements, see [bfd-liveness-detection](#).



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

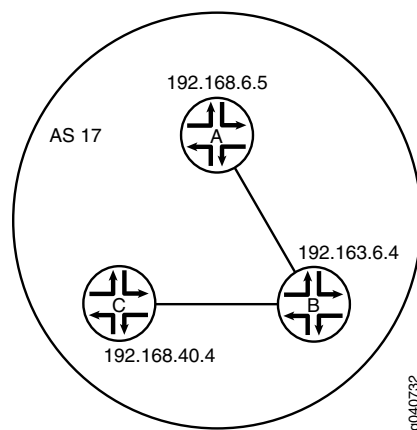
Depending on your network environment, these additional recommendations might apply:

- 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 NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

BFD is supported on the default routing instance (the main router), routing instances, and logical systems. This example shows BFD on logical systems.

Figure 42 on page 351 shows a typical network with internal peer sessions.

**Figure 42: Typical Network with IBGP Sessions**



### 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 A**     **set logical-systems A interfaces lt-1/2/0 unit 1 description to-B**  
                  **set logical-systems A interfaces lt-1/2/0 unit 1 encapsulation ethernet**  
                  **set logical-systems A interfaces lt-1/2/0 unit 1 peer-unit 2**

```

set logical-systems A interfaces lt-1/2/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-bfd
set logical-systems A protocols bgp group internal-peers traceoptions flag bfd detail
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17

```

**Device B**

```

set logical-systems B interfaces lt-1/2/0 unit 2 description to-A
set logical-systems B interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-1/2/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-1/2/0 unit 5 description to-C
set logical-systems B interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-1/2/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17

```

**Device C**

```

set logical-systems C interfaces lt-1/2/0 unit 6 description to-B
set logical-systems C interfaces lt-1/2/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-1/2/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-1/2/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4

```

```

set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-1/2/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol
  direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17

```

### Configuring Device A

**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 Device A:

1. Set the CLI to Logical System A.

```
user@host> set cli logical-system A
```

2. Configure the interfaces.

```

[edit interfaces lt-1/2/0 unit 1]
user@host:A# set description to-B
user@host:A# set encapsulation ethernet
user@host:A# set peer-unit 2
user@host:A# set family inet address 10.10.10.1/30

```

```

[edit interfaces lo0 unit 1]
user@host:A# set family inet address 192.168.6.5/32

```

3. Configure BGP.

The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```

[edit protocols bgp group internal-peers]
user@host:A# set type internal
user@host:A# set local-address 192.168.6.5
user@host:A# set export send-direct
user@host:A# set neighbor 192.163.6.4
user@host:A# set neighbor 192.168.40.4

```

4. Configure BFD.

```

[edit protocols bgp group internal-peers]
user@host:A# set bfd-liveness-detection minimum-interval 1000

```

You must configure the same minimum interval on the connecting peer.

5. (Optional) Configure BFD tracing.

```

[edit protocols bgp group internal-peers]
user@host:A# set traceoptions file bgp-bfd
user@host:A# set traceoptions flag bfd detail

```

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
```

```

user@host:A# set interface lo0.1 passive
user@host:A# set interface lt-1/2/0.1

```

7. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```

[edit policy-options policy-statement send-direct term 2]
user@host:A# set from protocol direct
user@host:A# set then accept

```

8. Configure the router ID and the autonomous system (AS) number.

```

[edit routing-options]
user@host:A# set router-id 192.168.6.5
user@host:A# set autonomous-system 17

```

9. If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps to configure Device B and Device C.

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

```

user@host:A# show interfaces
lt-1/2/0 {
  unit 1 {
    description to-B;
    encapsulation ethernet;
    peer-unit 2;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}

user@host:A# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}

user@host:A# show protocols
bgp {
  group internal-peers {
    type internal;
    traceoptions {

```

```

        file bgp-bfd;
        flag bfd detail;
    }
    local-address 192.168.6.5;
    export send-direct;
    bfd-liveness-detection {
        minimum-interval 1000;
    }
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
}
}
ospf {
    area 0.0.0.0 {
        interface lo0.1 {
            passive;
        }
        interface lt-1/2/0.1;
    }
}

user@host:A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;

```

### Verification

Confirm that the configuration is working properly.

- [Verifying That BFD Is Enabled on page 355](#)
- [Verifying That BFD Sessions Are Up on page 356](#)
- [Viewing Detailed BFD Events on page 356](#)
- [Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface on page 357](#)

#### *Verifying That BFD Is Enabled*

<b>Purpose</b>	Verify that BFD is enabled between the IBGP peers.
<b>Action</b>	<p>From operational mode, enter the <b>show bgp neighbor</b> command. You can use the <b>  match bfd</b> filter to narrow the output.</p> <pre> user@host:A&gt; show bgp neighbor   match bfd Options: &lt;BfdEnabled&gt; BFD: enabled, up Trace file: /var/log/A/bgp-bfd size 131072 files 10 Options: &lt;BfdEnabled&gt; BFD: enabled, up Trace file: /var/log/A/bgp-bfd size 131072 files 10 </pre>
<b>Meaning</b>	The output shows that Logical System A has two neighbors with BFD enabled. When BFD is not enabled, the output displays <b>BFD: disabled, down</b> , and the <b>&lt;BfdEnabled&gt;</b> option is absent. If BFD is enabled and the session is down, the output displays <b>BFD: enabled</b> ,

**down.** The output also shows that BFD-related events are being written to a log file because trace operations are configured.

### *Verifying That BFD Sessions Are Up*

**Purpose** Verify that the BFD sessions are up, and view details about the BFD sessions.

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

```
user@host:A> show bfd session extensive
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
192.163.6.4	Up		3.000	1.000	3
Client BGP, TX interval 1.000, RX interval 1.000					
Session up time 00:54:40					
Local diagnostic None, remote diagnostic None					
Remote state Up, version 1					
Logical system 12, routing table index 25					
Min async interval 1.000, min slow interval 1.000					
Adaptive async TX interval 1.000, RX interval 1.000					
Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3					
Remote min TX interval 1.000, min RX interval 1.000, multiplier 3					
Local discriminator 10, remote discriminator 9					
Echo mode disabled/inactive					
Multi-hop route table 25, local-address 192.168.6.5					

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
192.168.40.4	Up		3.000	1.000	3
Client BGP, TX interval 1.000, RX interval 1.000					
Session up time 00:48:03					
Local diagnostic None, remote diagnostic None					
Remote state Up, version 1					
Logical system 12, routing table index 25					
Min async interval 1.000, min slow interval 1.000					
Adaptive async TX interval 1.000, RX interval 1.000					
Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3					
Remote min TX interval 1.000, min RX interval 1.000, multiplier 3					
Local discriminator 14, remote discriminator 13					
Echo mode disabled/inactive					
Multi-hop route table 25, local-address 192.168.6.5					

2 sessions, 2 clients

Cumulative transmit rate 2.0 pps, cumulative receive rate 2.0 pps

**Meaning** The TX interval 1.000, RX interval 1.000 output represents the setting configured with the **minimum-interval** statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the **bfd-liveness-detection** statement.

### *Viewing Detailed BFD Events*

**Purpose** View the contents of the BFD trace file to assist in troubleshooting, if needed.

**Action** From operational mode, enter the file `show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
```

```

Aug 15 17:07:25 trace_on: Tracing to "/var/log/A/bgp-bfd" started
Aug 15 17:07:26.492190 bgp_peer_init: BGP peer 192.163.6.4 (Internal AS 17) local
address 192.168.6.5 not found. Leaving peer idled
Aug 15 17:07:26.493176 bgp_peer_init: BGP peer 192.168.40.4 (Internal AS 17) local
address 192.168.6.5 not found. Leaving peer idled
Aug 15 17:07:32.597979 task_connect: task BGP_17.192.163.6.4+179 addr
192.163.6.4+179: No route to host
Aug 15 17:07:32.599623 bgp_connect_start: connect 192.163.6.4 (Internal AS 17):
No route to host
Aug 15 17:07:36.869394 task_connect: task BGP_17.192.168.40.4+179 addr
192.168.40.4+179: No route to host
Aug 15 17:07:36.870624 bgp_connect_start: connect 192.168.40.4 (Internal AS 17):
No route to host
Aug 15 17:08:04.599220 task_connect: task BGP_17.192.163.6.4+179 addr
192.163.6.4+179: No route to host
Aug 15 17:08:04.601135 bgp_connect_start: connect 192.163.6.4 (Internal AS 17):
No route to host
Aug 15 17:08:08.869717 task_connect: task BGP_17.192.168.40.4+179 addr
192.168.40.4+179: No route to host
Aug 15 17:08:08.869934 bgp_connect_start: connect 192.168.40.4 (Internal AS 17):
No route to host
Aug 15 17:08:36.603544 advertising receiving-speaker only capability to neighbor
192.163.6.4 (Internal AS 17)
Aug 15 17:08:36.606726 bgp_read_message: 192.163.6.4 (Internal AS 17): 0 bytes
buffered
Aug 15 17:08:36.609119 Initiated BFD session to peer 192.163.6.4 (Internal AS
17): address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3
ver=255
Aug 15 17:08:36.734033 advertising receiving-speaker only capability to neighbor
192.168.40.4 (Internal AS 17)
Aug 15 17:08:36.738436 Initiated BFD session to peer 192.168.40.4 (Internal AS
17): address=192.168.40.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3
ver=255
Aug 15 17:08:40.537552 BFD session to peer 192.163.6.4 (Internal AS 17) up
Aug 15 17:08:40.694410 BFD session to peer 192.168.40.4 (Internal AS 17) up

```

**Meaning** Before the routes are established, the **No route to host** message appears in the output. After the routes are established, the last two lines show that both BFD sessions come up.

#### *Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface*

**Purpose** Check to see what happens after bringing down a router or switch and then bringing it back up. To simulate bringing down a router or switch, deactivate the loopback interface on Logical System B.

**Action** 1. From configuration mode, enter the **deactivate logical-systems B interfaces lo0 unit 2 family inet** command.

```

user@host:A# deactivate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit

```

2. From operational mode, enter the **file show /var/log/A/bgp-bfd** command.

```

user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:20:55.995648 bgp_read_v4_message:9747: NOTIFICATION received from
192.163.6.4 (Internal AS 17): code 6 (Cease) subcode 6 (Other Configuration
Change)

```

```
Aug 15 17:20:56.004508 Terminated BFD session to peer 192.163.6.4 (Internal AS 17)
Aug 15 17:21:28.007755 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No route to host
Aug 15 17:21:28.008597 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to host
```

3. From configuration mode, enter the **activate logical-systems B interfaces lo0 unit 2 family inet** command.

```
user@host:A# activate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

4. From operational mode, enter the **file show /var/log/A/bgp-bfd** command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:25:53.623743 advertising receiving-speaker only capability to neighbor 192.163.6.4 (Internal AS 17)
Aug 15 17:25:53.631314 Initiated BFD session to peer 192.163.6.4 (Internal AS 17): address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:25:57.570932 BFD session to peer 192.163.6.4 (Internal AS 17) up
```

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

---

## Example: Configuring BFD Authentication for BGP

---

- [Understanding BFD Authentication for BGP on page 358](#)
- [Example: Configuring BFD Authentication for BGP on page 360](#)

### Understanding BFD Authentication for BGP

Bidirectional Forwarding Detection protocol (BFD) enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when you run 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 BGP. BFD authentication is not supported on MPLS OAM sessions. BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

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 level of authentication that can be configured:

- [BFD Authentication Algorithms on page 359](#)
- [Security Authentication Keychains on page 360](#)
- [Strict Versus Loose Authentication on page 360](#)

### 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 (NSR) 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 BGP

Beginning with Junos OS Release 9.6, you can configure authentication for BFD sessions running over BGP. Only three steps are needed to configure authentication on a BFD session:

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

The following sections provide instructions for configuring and viewing BFD authentication on BGP:

- [Configuring BFD Authentication Parameters on page 360](#)
- [Viewing Authentication Information for BFD Sessions on page 362](#)

### Configuring BFD Authentication Parameters

---

BFD authentication can be configured for the entire BGP protocol, or a specific BGP group, neighbor, or routing instance.

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 BFD authentication:

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

```
[edit]
user@host# set protocols bgp bfd-liveness-detection authentication algorithm
keyed-sha-1
user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
algorithm keyed-sha-1
user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
authentication algorithm keyed-sha-1
```



**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.

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

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

```
[edit]
user@host# set protocols bgp bfd-liveness-detection authentication keychain bfd-bgp
user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
keychain bfd-bgp
user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
authentication keychain bfd-bgp
```



**NOTE:** 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.

3. Specify the unique security authentication information for BFD sessions:
  - The matching keychain name as specified in Step 2.
  - 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]
```

```
user@host# set authentication-key-chains key-chain bfd-bgp key 53 secret
$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm start-time 2009-06-14.10:00:00
```

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

```
[edit]
user@host# set protocols bgp bfd-liveness-detection authentication loose-check
user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
loose-check
user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
authentication loose-check
```

- (Optional) View your configuration using the **show bfd session detail** or **show bfd session extensive** command.
- Repeat these steps to configure the other end of the BFD session.



**NOTE:** BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

### Viewing Authentication Information for BFD Sessions

You can view the existing BFD authentication configuration using the **show bfd session detail** and **show bfd session extensive** commands.

The following example shows BFD authentication configured for the **bgp-gr1** BGP group. It specifies the keyed SHA-1 authentication algorithm and a keychain name of **bfd-bgp**. The authentication keychain is configured with two keys. Key 1 contains the secret data “\$9\$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm” and a start time of June 1, 2009, at 9:46:02 AM PST. Key 2 contains the secret data “\$9\$a5jiKW9l.reP38ny.TszF2/9” and a start time of June 1, 2009, at 3:29:20 PM PST.

```
[edit protocols bgp]
group bgp-gr1 {
  bfd-liveness-detection {
    authentication {
      algorithm keyed-sha-1;
      key-chain bfd-bgp;
    }
  }
}
[edit security]
authentication key-chains {
  key-chain bfd-bgp {
    key 1 {
      secret "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm";
      start-time "2009-6-1.09:46:02 -0700";
    }
    key 2 {
      secret "$9$a5jiKW9l.reP38ny.TszF2/9";
      start-time "2009-6-1.15:29:20 -0700";
    }
  }
}
```

```
}
```

If you commit these updates to your configuration, you see output similar to the following. In the output for the **show bfd session detail** command, **Authenticate** is displayed to indicate that BFD authentication is configured. For more information about the configuration, use the **show bfd session extensive** command. The output for this command provides the keychain name, the authentication algorithm and mode for each client in the session, and the overall BFD authentication configuration status, keychain name, and authentication algorithm and mode.

#### show bfd session detail

```
user@host# show bfd session detail
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
50.0.0.2	Up	ge-0/1/5.0	0.900	0.300	3

Client BGP, TX interval 0.300, RX interval 0.300, **Authenticate**  
 Session up time 3d 00:34  
 Local diagnostic None, remote diagnostic NbrSignal  
 Remote state Up, version 1  
 Replicated

#### show bfd session extensive

```
user@host# show bfd session extensive
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
50.0.0.2	Up	ge-0/1/5.0	0.900	0.300	3

Client BGP, TX interval 0.300, RX interval 0.300, **Authenticate**  
**keychain bfd-bgp, algo keyed-sha-1, mode strict**  
 Session up time 00:04:42  
 Local diagnostic None, remote diagnostic NbrSignal  
 Remote state Up, version 1  
 Replicated  
 Min async interval 0.300, min slow interval 1.000  
 Adaptive async TX interval 0.300, RX interval 0.300  
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3  
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3  
 Local discriminator 2, remote discriminator 2  
 Echo mode disabled/inactive  
**Authentication enabled/active, keychain bfd-bgp, algo keyed-sha-1, mode strict**

- Related Documentation**
- [Understanding External BGP Peering Sessions on page 33](#)
  - [BGP Configuration Overview on page 12](#)



## CHAPTER 8

# BGP Load Balancing Configuration

- [Examples: Configuring BGP Multipath on page 365](#)
- [Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths on page 382](#)
- [Example: Advertising Multiple BGP Paths to a Destination on page 392](#)

### Examples: Configuring BGP Multipath

---

- [Understanding BGP Multipath on page 365](#)
- [Example: Load Balancing BGP Traffic on page 366](#)
- [Example: Configuring Single-Hop EBGP Peers to Accept Remote Next Hops on page 371](#)

### Understanding BGP Multipath

BGP multipath allows you to select multiple internal or external BGP peers as active paths. Selecting multiple paths enables BGP peering to load-balance traffic across an Autonomous System (AS) confederation boundary.

A path is considered a BGP equal-cost path (and is used for forwarding) if a tie-break is performed. The tie-break is performed after the BGP route path selection step that chooses the next-hop path that is resolved through the IGP route with the lowest metric. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor are considered in the path selection process.

BGP, typically selects only one best path for each prefix and installs that route in the routing table. When BGP multipath is enabled, the device selects multiple equal-cost EBGP paths as the best paths to reach a given destination, and all these paths are installed in the routing table. BGP advertises only the active path to its neighbors. However, you can configure BGP to advertise multiple paths to the same destination for redundancy and load balancing.

The Junos OS BGP multipath feature supports the following applications:

- Load balancing across multiple links between two routing devices belonging to different autonomous systems (ASs)
- Load balancing across a common subnet or multiple subnets to different routing devices belonging to the same peer AS

- Load balancing across multiple links between two routing devices belonging to different external confederation peers
- Load balancing across a common subnet or multiple subnets to different routing devices belonging to external confederation peers

In a common scenario for load balancing, a customer is multihomed to multiple routers in a point of presence (POP). The default behavior is to send all traffic across only one of the available links. Load balancing causes traffic to use two or more of the links.



**NOTE:** BGP multipath does not apply to paths that share the same MED-plus-IGP cost, yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

## Example: Load Balancing BGP Traffic

This example shows how to configure BGP to select multiple equal-cost external BGP (EBGP) or internal BGP (IBGP) paths as active paths.

- [Requirements on page 366](#)
- [Overview on page 366](#)
- [Configuration on page 367](#)
- [Verification on page 369](#)

### Requirements

---

Before you begin:

- Configure the device interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure a routing policy that exports routes (such as direct routes or IGP routes) from the routing table into BGP.

### Overview

---

The following steps show how to configure per-packet load balancing:

1. Define a load-balancing routing policy by including one or more **policy-statement** statements at the **[edit policy-options]** hierarchy level, defining an action of **load-balance per-packet**:

```
policy-statement policy-name {  
  from {  
    match-conditions;  
    route-filter destination-prefix match-type <actions>;  
    prefix-list name;  
  }  
  then {
```

```

        load-balance per-packet;
    }
}

```

2. Apply the policy to routes exported from the routing table to the forwarding table. To do this, include the **forwarding-table** and **export** statements:

```

forwarding-table {
    export policy-name;
}

```

You cannot apply the export policy to VRF routing instances.

3. Specify all next hops of that route, if more than one exists, when allocating a label corresponding to a route that is being advertised.
4. Configure the forwarding-options hash key for MPLS to include the IP payload.



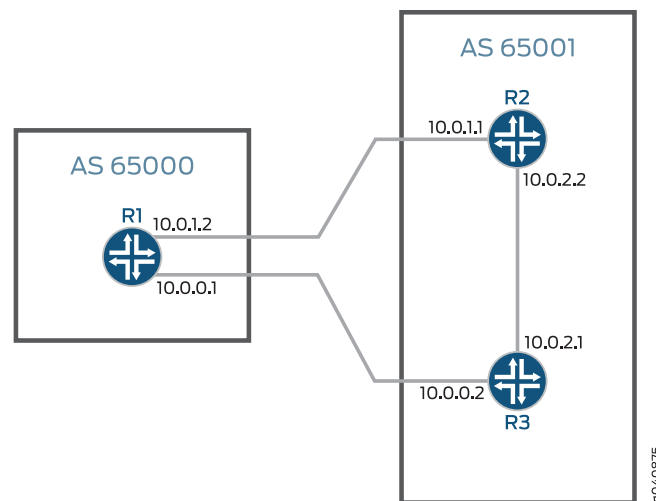
**NOTE:** On some platforms, you can increase the number of paths that are load balanced by using the **chassis maximum-ecmp** statement. With this statement, you can change the maximum number of equal-cost load-balanced paths to 32 or 64.

In this example, Device R1 is in AS 65000 and is connected to both Device R2 and Device R3, which are in AS 65001. This example shows the configuration on Device R1.

### Topology

Figure 43 on page 367 shows the topology used in this example.

Figure 43: BGP Load Balancing



### 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.

```
set protocols bgp group external type external
set protocols bgp group external peer-as 65001
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.1
set protocols bgp group external neighbor 10.0.0.2
set policy-options policy-statement loadbal from route-filter 10.0.0.0/16 orlonger
set policy-options policy-statement loadbal then load-balance per-packet
set routing-options forwarding-table export loadbal
set routing-options autonomous-system 65000
```

**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 BGP peer sessions:

1. Configure the BGP group.

```
[edit protocols bgp group external]
user@R1# set type external
user@R1# set peer-as 65001
user@R1# set neighbor 10.0.1.1
user@R1# set neighbor 10.0.0.2
```

2. Enable the BGP group to use multiple paths.



**NOTE:** To disable the default check requiring that paths accepted by BGP multipath must have the same neighboring autonomous system (AS), include the `multiple-as` option.

```
[edit protocols bgp group external]
user@R1# set multipath
```

3. Configure the load-balancing policy.

```
[edit policy-options policy-statement loadbal]
user@R1# set from route-filter 10.0.0.0/16 orlonger
user@R1# set then load-balance per-packet
```

4. Apply the load-balancing policy.

```
[edit routing-options]
user@R1# set forwarding-table export loadbal
```

5. Configure the local autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

**Results** From configuration mode, confirm your configuration by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display

the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@R1# show protocols
bgp {
  group external {
    type external;
    peer-as 65001;
    multipath;
    neighbor 10.0.1.1;
    neighbor 10.0.0.2;
  }
}

[edit]
user@R1# show policy-options
policy-statement loadbal {
  from {
    route-filter 10.0.0.0/16 orlonger;
  }
  then {
    load-balance per-packet;
  }
}

[edit]
user@R1# show routing-options
autonomous-system 65000;
forwarding-table {
  export loadbal;
}
```

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

## Verification

Confirm that the configuration is working properly:

- [Verifying Routes on page 369](#)
- [Verifying Forwarding on page 371](#)

### Verifying Routes

**Purpose** Verify that routes are learned from both routers in the neighboring AS.

**Action** From operational mode, run the **show route** command.

```
user@R1> show route 10.0.2.0
inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.2.0/30          * [BGP/170] 03:12:32, localpref 100
                    AS path: 65001 I
                    to 10.0.1.1 via ge-1/2/0.0
                    > to 10.0.0.2 via ge-1/2/1.0
                    [BGP/170] 03:12:32, localpref 100
```

```
AS path: 65001 I
> to 10.0.1.1 via ge-1/2/0.0
```

```
user@R1> show route 10.0.2.0 detail
inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
10.0.2.0/30 (2 entries, 1 announced)
  *BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 262142
        Next-hop reference count: 3
        Source: 10.0.0.2
        Next hop: 10.0.1.1 via ge-1/2/0.0
        Next hop: 10.0.0.2 via ge-1/2/1.0, selected
        State: <Active Ext>
        Local AS: 65000 Peer AS: 65001
        Age: 3:18:30
        Task: BGP_65001.10.0.0.2+55402
        Announcement bits (1): 2-KRT
        AS path: 65001 I
        Accepted Multipath
        Localpref: 100
        Router ID: 192.168.2.1
  BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 602
        Next-hop reference count: 5
        Source: 10.0.1.1
        Next hop: 10.0.1.1 via ge-1/2/0.0, selected
        State: <NotBest Ext>
        Inactive reason: Not Best in its group - Active preferred
        Local AS: 65000 Peer AS: 65001
        Age: 3:18:30
        Task: BGP_65001.10.0.1.1+53135
        AS path: 65001 I
        Accepted
        Localpref: 100
        Router ID: 192.168.3.1
```

**Meaning** The active path, denoted with an asterisk (\*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 10.0.2.0 destination. The 10.0.1.1 next hop is copied from the inactive path to the active path.



**NOTE:** The `show route detail` command output designates one gateway as selected. This output is potentially confusing in the context of load balancing. The selected gateway is used for many purposes in addition to deciding which gateway to install into the kernel when Junos OS is not performing per-packet load-balancing. For instance, the `ping mpls` command uses the selected gateway when sending packets. Multicast protocols use the selected gateway in some cases to determine the upstream interface. Therefore, even when Junos OS is performing per-packet load-balancing by way of a forwarding-table policy, the selected gateway information is still required for other purposes. It is useful to display the selected gateway for troubleshooting purposes. Additionally, it is possible to use forwarding-table policy to override what is installed into the kernel (for example, by using the `install-nexthop` action). In this case, the next-hop gateway installed in the forwarding table might be a subset of the total gateways displayed in the `show route` command.

### Verifying Forwarding

**Purpose** Verify that both next hops are installed in the forwarding table.

**Action** From operational mode, run the **show route forwarding-table** command.

```
user@R1> show route forwarding-table destination 10.0.2.0
```

```
Routing table: default.inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
10.0.2.0/30	user	0		ulst	262142	2	
			10.0.1.1	ucst	602	5	ge-1/2/0.0
			10.0.0.2	ucst	522	6	ge-1/2/1.0

### Example: Configuring Single-Hop EBGPeers to Accept Remote Next Hops

This example shows how to configure a single-hop external BGP (EBGP) peer to accept a remote next hop with which it does not share a common subnet.

- [Requirements on page 371](#)
- [Overview on page 371](#)
- [Configuration on page 372](#)
- [Verification on page 380](#)

#### Requirements

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

#### Overview

In some situations, it is necessary to configure a single-hop EBGPeer to accept a remote next hop with which it does not share a common subnet. The default behavior is for any next-hop address received from a single-hop EBGPeer that is not recognized as sharing a common subnet to be discarded. The ability to have a single-hop EBGPeer accept a remote next hop to which it is not directly connected also prevents you from having to configure the single-hop EBGPeer neighbor as a multihop session. When you configure a multihop session in this situation, all next-hop routes learned through this EBGPeer are labeled indirect even when they do share a common subnet. This situation breaks multipath functionality for routes that are recursively resolved over routes that include these next-hop addresses. Configuring the **accept-remote-nexthop** statement allows a single-hop EBGPeer to accept a remote next hop, which restores multipath functionality for routes that are resolved over these next-hop addresses. You can configure this statement at the global, group, and neighbor hierarchy levels for BGP. The statement is also supported on logical systems and the VPN routing and forwarding (VRF) routing instance type. Both the remote next-hop and the EBGPeer must support BGP route refresh as defined in RFC 2918, *Route Refresh Capability in BGP-4*. If the remote peer does not support BGP route refresh, the session is reset.



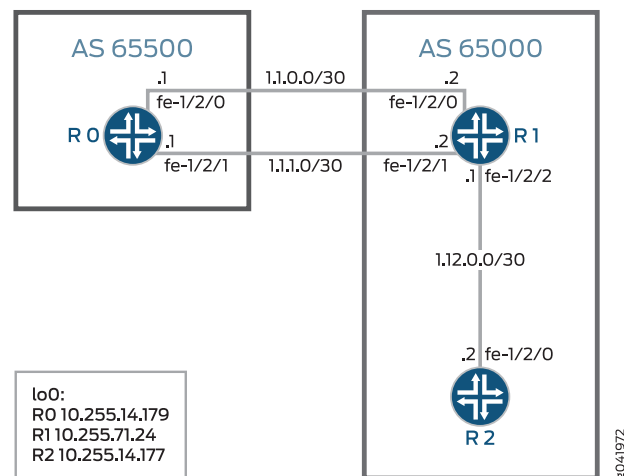
**NOTE:** You cannot configure both the `multihop` and `accept-remote-nexthop` statements for the same EBGP peer.

When you enable a single-hop EBGP peer to accept a remote next hop, you must also configure an import routing policy on the EBGP peer that specifies the remote next-hop address.

This example includes an import routing policy, `agg_route`, that enables a single-hop external BGP peer (Device R1) to accept the remote next-hop 1.1.10.10 for the route to the 1.1.230.0/23 network. At the `[edit protocols bgp]` hierarchy level, the example includes the `import agg_route` statement to apply the policy to the external BGP peer and includes the `accept-remote-nexthop` statement to enable the single-hop EBGP peer to accept the remote next hop.

Figure 44 on page 372 shows the sample topology.

**Figure 44: Topology for Accepting a Remote Next Hop**



### Configuration

- [Device R0 on page 373](#)
- [Configuring Device R1 on page 376](#)
- [Configuring Device R2 on page 378](#)

#### 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 R0
set interfaces fe-1/2/0 unit 1 family inet address 1.1.0.1/30
set interfaces fe-1/2/1 unit 2 family inet address 1.1.1.1/30
set interfaces lo0 unit 1 family inet address 10.255.14.179/32
set protocols bgp group ext type external
set protocols bgp group ext export test_route
set protocols bgp group ext export agg_route
```

```

set protocols bgp group ext peer-as 65000
set protocols bgp group ext multipath
set protocols bgp group ext neighbor 1.1.0.2
set protocols bgp group ext neighbor 1.1.1.2
set policy-options policy-statement agg_route term 1 from protocol static
set policy-options policy-statement agg_route term 1 from route-filter 1.1.230.0/23 exact
set policy-options policy-statement agg_route term 1 then accept
set policy-options policy-statement test_route term 1 from protocol static
set policy-options policy-statement test_route term 1 from route-filter 1.1.10.10/32 exact
set policy-options policy-statement test_route term 1 then accept
set routing-options static route 1.1.10.10/32 reject
set routing-options static route 1.1.230.0/23 reject
set routing-options autonomous-system 65500

```

**Device R1**

```

set interfaces fe-1/2/0 unit 3 family inet address 1.1.0.2/30
set interfaces fe-1/2/1 unit 4 family inet address 1.12.0.1/30
set interfaces fe-1/2/2 unit 5 family inet address 1.1.1.2/30
set interfaces lo0 unit 2 family inet address 10.255.71.24/32
set protocols bgp accept-remote-nexthop
set protocols bgp group ext type external
set protocols bgp group ext import agg_route
set protocols bgp group ext peer-as 65500
set protocols bgp group ext multipath
set protocols bgp group ext neighbor 1.1.0.1
set protocols bgp group ext neighbor 1.1.1.1
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.71.24
set protocols bgp group int neighbor 10.255.14.177
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set protocols ospf area 0.0.0.0 interface 10.255.71.24
set policy-options policy-statement agg_route term 1 from protocol bgp
set policy-options policy-statement agg_route term 1 from route-filter 1.1.230.0/23 exact
set policy-options policy-statement agg_route term 1 then next-hop 1.1.10.10
set policy-options policy-statement agg_route term 1 then accept
set routing-options autonomous-system 65000

```

**Device R2**

```

set interfaces fe-1/2/0 unit 6 family inet address 1.12.0.2/30
set interfaces lo0 unit 3 family inet address 10.255.14.177/32
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.177
set protocols bgp group int neighbor 10.255.71.24
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6
set protocols ospf area 0.0.0.0 interface 10.255.14.177
set routing-options autonomous-system 65000

```

### **Device R0**

**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 Device R0:

1. Configure the interfaces.  
[edit interfaces fe-1/2/0 unit 1]

```
user@R0# set family inet address 1.1.0.1/30
```

```
[edit interfaces fe-1/2/1 unit 2]
```

```
user@R0# set family inet address 1.1.1.1/30
```

```
[edit interfaces lo0 unit 1]
```

```
user@R0# set family inet address 10.255.14.179/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
```

```
user@R0# set type external
```

```
user@R0# set peer-as 65000
```

```
user@R0# set neighbor 1.1.0.2
```

```
user@R0# set neighbor 1.1.1.2
```

3. Enable multipath BGP between Device R0 and Device R1.

```
[edit protocols bgp group ext]
```

```
user@R0# set multipath
```

4. Configure static routes to remote networks.

These routes are not part of the topology. The purpose of these routes is to demonstrate the functionality in this example.

```
[edit routing-options]
```

```
user@R0# set static route 1.1.10.10/32 reject
```

```
user@R0# set static route 1.1.230.0/23 reject
```

5. Configure routing policies that accept the static routes.

```
[edit policy-options policy-statement agg_route term 1]
```

```
user@R0# set from protocol static
```

```
user@R0# set from route-filter 1.1.230.0/23 exact
```

```
user@R0# set then accept
```

```
[edit policy-options policy-statement test_route term 1]
```

```
user@R0# set from protocol static
```

```
user@R0# set from route-filter 1.1.10.10/32 exact
```

```
user@R0# set then accept
```

6. Export the **agg\_route** and **test\_route** policies from the routing table into BGP.

```
[edit protocols bgp group ext]
```

```
user@R0# set export test_route
```

```
user@R0# set export agg_route
```

7. Configure the autonomous system (AS) number.

```
[edit routing-options]
```

```
user@R0# set autonomous-system 65500
```

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

```
user@R0# show interfaces
```

```

fe-1/2/0 {
  unit 1 {
    family inet {
      address 1.1.0.1/30;
    }
  }
}
fe-1/2/1 {
  unit 2 {
    family inet {
      address 1.1.1.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.255.14.179/32;
    }
  }
}

user@R0# show policy-options
policy-statement agg_route {
  term 1 {
    from {
      protocol static;
      route-filter 1.1.230.0/23 exact;
    }
    then accept;
  }
}
policy-statement test_route {
  term 1 {
    from {
      protocol static;
      route-filter 1.1.10.10/32 exact;
    }
    then accept;
  }
}

user@R0# show protocols
bgp {
  group ext {
    type external;
    export [ test_route agg_route ];
    peer-as 65000;
    multipath;
    neighbor 1.1.0.2;
    neighbor 1.1.1.2;
  }
}

user@R0# show routing-options
static {
  route 1.1.10.10/32 reject;
}

```

```
route 1.1.230.0/23 reject;  
}  
autonomous-system 65500;
```

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

### **Configuring Device R1**

**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 Device R1:

1. Configure the interfaces.  

```
[edit interfaces fe-1/2/0 unit 3]  
user@R1# set family inet address 1.1.0.2/30  
  
[edit interfaces fe-1/2/1 unit 4]  
user@R1# set family inet address 1.12.0.1/30  
  
[edit interfaces fe-1/2/2 unit 5]  
user@R1# set family inet address 1.1.1.2/30  
  
[edit interfaces lo0 unit 2]  
user@R1# set family inet address 10.255.71.24/32
```
2. Configure OSPF.  

```
[edit protocols ospf area 0.0.0.0]  
user@R1# set interface fe-1/2/1.4  
user@R1# set interface 10.255.71.24
```
3. Enable Device R1 to accept the remote next hop.  

```
[edit protocols bgp]  
user@R1# set accept-remote-nexthop
```
4. Configure IBGP.  

```
[edit protocols bgp group int]  
user@R1# set type internal  
user@R1# set local-address 10.255.71.24  
user@R1# set neighbor 10.255.14.177
```
5. Configure EBGP.  

```
[edit protocols bgp group ext]  
user@R1# set type external  
user@R1# set peer-as 65500  
user@R1# set neighbor 1.1.0.1  
user@R1# set neighbor 1.1.1.1
```
6. Enable multipath BGP between Device R0 and Device R1.  

```
[edit protocols bgp group ext]  
user@R1# set multipath
```

7. Configure a routing policy that enables a single-hop external BGP peer (Device R1) to accept the remote next-hop 1.1.10.10 for the route to the 1.1.230.0/23 network.

```
[edit policy-options policy-statement agg_route term 1]
user@R1# set from protocol bgp
user@R1# set from route-filter 1.1.230.0/23 exact
user@R1# set then next-hop 1.1.10.10
user@R1# set then accept
```

8. Import the **agg\_route** policy into the routing table on Device R1.

```
[edit protocols bgp group ext]
user@R1# set import agg_route
```

9. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 1.1.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 4 {
    family inet {
      address 1.12.0.1/30;
    }
  }
}
fe-1/2/2 {
  unit 5 {
    family inet {
      address 1.1.1.2/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 10.255.71.24/32;
    }
  }
}
```

```
user@R1# show policy-options
policy-statement agg_route {
  term 1 {
```

```

    from {
        protocol bgp;
        route-filter 1.1.230.0/23 exact;
    }
    then {
        next-hop 1.1.10.10;
        accept;
    }
}
}

```

```

user@R1# show protocols
bgp {
    accept-remote-nexthop;
    group ext {
        type external;
        import agg_route;
        peer-as 65500;
        multipath;
        neighbor 1.1.0.1;
        neighbor 1.1.1.1;
    }
    group int {
        type internal;
        local-address 10.255.71.24;
        neighbor 10.255.14.177;
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.4;
        interface 10.255.71.24;
    }
}

```

```

user@R1# show routing-options
autonomous-system 65000;

```

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

### Configuring Device R2

**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 Device R2:

1. Configure the interfaces.
 

```

[edit interfaces fe-1/2/0 unit 6]
user@R2# set family inet address 1.12.0.2/30

[edit interfaces lo0 unit 3]
user@R2# set family inet address 10.255.14.177/32

```
2. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface fe-1/2/0.6
user@R2# set interface 10.255.14.177
```

3. Configure IBGP.

```
[edit protocols bgp group int]
user@R2# set type internal
user@R2# set local-address 10.255.14.177
user@R2# set neighbor 10.255.71.24
```

4. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 6 {
    family inet {
      address 1.12.0.2/30;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 10.255.14.177/32;
    }
  }
}

user@R2# show protocols
bgp {
  group int {
    type internal;
    local-address 10.255.14.177;
    neighbor 10.255.71.24;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.6;
    interface 10.255.14.177;
  }
}

user@R2# show routing-options
autonomous-system 65000;
```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying That the Multipath Route with the Indirect Next Hop Is in the Routing Table on page 380](#)
- [Deactivating and Reactivating the accept-remote-nexthop Statement on page 381](#)

### *Verifying That the Multipath Route with the Indirect Next Hop Is in the Routing Table*

**Purpose** Verify that Device R1 has a route to the 1.1.230.0/23 network.

**Action** From operational mode, enter the **show route 1.1.230.0 extensive** command.

```

user@R1> show route 1.1.230.0 extensive
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
Restart Complete
1.1.230.0/23 (2 entries, 1 announced)
TSI:
KRT in-kernel 1.1.230.0/23 -> {indirect(262142)}
Page 0 idx 1 Type 1 val 9168f6c
  Nexthop: 1.1.10.10
  Localpref: 100
  AS path: [65000] 65500 I
  Communities:
Path 1.1.230.0 from 1.1.0.1 Vector len 4. Val: 1
  *BGP Preference: 170/-101
    Next hop type: Indirect
    Address: 0x90c44d8
    Next-hop reference count: 4
    Source: 1.1.0.1
    Next hop type: Router, Next hop index: 262143
    Next hop: 1.1.0.1 via fe-1/2/0.3, selected
    Next hop: 1.1.1.1 via fe-1/2/2.5
    Protocol next hop: 1.1.10.10
    Indirect next hop: 91c0000 262142
    State: <Active Ext>
    Local AS: 65000 Peer AS: 65500
    Age: 2:55:31 Metric2: 0
    Task: BGP_65500.1.1.0.1+64631
    Announcement bits (3): 2-KRT 3-BGP_RT_Background 4-Resolve tree
1
    AS path: 65500 I
    Accepted Multipath
    Localpref: 100
    Router ID: 10.255.14.179
    Indirect next hops: 1
      Protocol next hop: 1.1.10.10
      Indirect next hop: 91c0000 262142
      Indirect path forwarding next hops: 2
        Next hop type: Router
        Next hop: 1.1.0.1 via fe-1/2/0.3
        Next hop: 1.1.1.1 via fe-1/2/2.5
      1.1.10.10/32 Originating RIB: inet.0
        Node path count: 1
        Forwarding nexthops: 2
          Nexthop: 1.1.0.1 via fe-1/2/0.3
          Nexthop: 1.1.1.1 via fe-1/2/2.5
  BGP Preference: 170/-101

```

```

Next hop type: Indirect
Address: 0x90c44d8
Next-hop reference count: 4
Source: 1.1.1.1
Next hop type: Router, Next hop index: 262143
Next hop: 1.1.0.1 via fe-1/2/0.3, selected
Next hop: 1.1.1.1 via fe-1/2/2.5
Protocol next hop: 1.1.10.10
Indirect next hop: 91c0000 262142
State: <NotBest Ext>
Inactive reason: Not Best in its group - Update source
Local AS: 65500 Peer AS: 65500
Age: 2:55:27 Metric2: 0
Task: BGP_65500.1.1.1.1+53260
AS path: 65500 I
Accepted
Localpref: 100
Router ID: 10.255.14.179
Indirect next hops: 1
  Protocol next hop: 1.1.10.10
  Indirect next hop: 91c0000 262142
  Indirect path forwarding next hops: 2
    Next hop type: Router
    Next hop: 1.1.0.1 via fe-1/2/0.3
    Next hop: 1.1.1.1 via fe-1/2/2.5
  1.1.10.10/32 Originating RIB: inet.0
  Node path count: 1
  Forwarding nexthops: 2
    Nexthop: 1.1.0.1 via fe-1/2/0.3
    Nexthop: 1.1.1.1 via fe-1/2/2.5

```

**Meaning** The output shows that Device R1 has a route to the 1.1.230.0 network with the multipath feature enabled (**Accepted Multipath**). The output also shows that the route has an indirect next hop of 1.1.10.10.

### *Deactivating and Reactivating the accept-remote-nexthop Statement*

**Purpose** Make sure that the multipath route with the indirect next hop is removed from the routing table when you deactivate the **accept-remote-nexthop** statement.

**Action** 1. From configuration mode, enter the **deactivate protocols bgp accept-remote-nexthop** command.

```

user@R1# deactivate protocols bgp accept-remote-nexthop
user@R1# commit

```

2. From operational mode, enter the **show route 1.1.230.0** command.

```

user@R1> show route 1.1.230.0

```

3. From configuration mode, reactivate the statement by entering the **activate protocols bgp accept-remote-nexthop** command.

```

user@R1# activate protocols bgp accept-remote-nexthop
user@R1# commit

```

4. From operational mode, reenter the **show route 1.1.230.0** command.

```

user@R1> show route 1.1.230.0

```

```

inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)

```

**Restart Complete**

+ = Active Route, - = Last Active, \* = Both

```

1.1.230.0/23      *[BGP/170] 03:13:19, localpref 100
                  AS path: 65500 I
                  > to 1.1.0.1 via fe-1/2/0.3
                  to 1.1.1.1 via fe-1/2/2.5
                  [BGP/170] 03:13:15, localpref 100, from 1.1.1.1
                  AS path: 65500 I
                  > to 1.1.0.1 via fe-1/2/0.3
                  to 1.1.1.1 via fe-1/2/2.5

```

**Meaning** When the **accept-remote-nexthop** statement is deactivated, the multipath route to the 1.1.230.0 network is removed from the routing table .

**Related Documentation**

- [Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers on page 274](#)
- [Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths on page 339](#)

## Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths

- [Understanding Load Balancing for BGP Traffic with Unequal Bandwidth Allocated to the Paths on page 382](#)
- [Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths on page 383](#)

## Understanding Load Balancing for BGP Traffic with Unequal Bandwidth Allocated to the Paths

The multipath option removes the tiebreakers from the active route decision process, thereby allowing otherwise equal cost BGP routes learned from multiple sources to be installed into the forwarding table. However, when the available paths are not equal cost, you may wish to load balance the traffic asymmetrically.

Once multiple next hops are installed in the forwarding table, a specific forwarding next hop is selected by the Junos OS per-prefix load-balancing algorithm. This process hashes against a packet's source and destination addresses to deterministically map the prefix pairing onto one of the available next hops. Per-prefix mapping works best when the hash function is presented with a large number of prefixes, such as might occur on an Internet peering exchange, and it serves to prevent packet reordering among pairs of communicating nodes.

An enterprise network normally wants to alter the default behavior to evoke a *per-packet* load-balancing algorithm. Per-packet is emphasized here because its use is a misnomer that stems from the historic behavior of the original Internet Processor ASIC. In reality, current Juniper Networks routers support per-prefix (default) and per-flow load balancing. The latter involves hashing against various Layer 3 and Layer 4 headers, including portions of the source address, destination address, transport protocol, incoming interface, and application ports. The effect is that now individual flows are hashed to a specific next

hop, resulting in a more even distribution across available next hops, especially when routing between fewer source and destination pairs.

With per-packet load balancing, packets comprising a communication stream between two endpoints might be resequenced, but packets within individual flows maintain correct sequencing. Whether you opt for per-prefix or per-packet load balancing, asymmetry of access links can present a technical challenge. Either way, the prefixes or flows that are mapped to, for example, a T1 link will exhibit degraded performance when compared to those flows that map to, for example, a Fast Ethernet access link. Worse yet, with heavy traffic loads, any attempt at equal load balancing is likely to result in total saturation of the T1 link and session disruption stemming from packet loss.

Fortunately, the Juniper Networks BGP implementation supports the notion of a bandwidth community. This extended community encodes the bandwidth of a given next hop, and when combined with multipath, the load-balancing algorithm distributes flows across the set of next hops proportional to their relative bandwidths. Put another way, if you have a 10-Mbps and a 1-Mbps next hop, on average nine flows will map to the high-speed next hop for every one that uses the low speed.

Use of BGP bandwidth community is supported only with per-packet load balancing.

The configuration task has two parts:

- Configure the external BGP (EBGP) peering sessions, enable multipath, and define an import policy to tag routes with a bandwidth community that reflects link speed.
- Enable per-packet (really per-flow) load balancing for optimal distribution of traffic.

### Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths

This example shows how to configure BGP to select multiple unequal-cost paths as active paths.

BGP communities can help you control routing policy. An example of a good use for BGP communities is unequal load balancing. When an autonomous system border router (ASBR) receives routes from directly connected external BGP (EBGP) neighbors, the ASBR then advertises those routes to internal neighbors, using IBGP advertisements. In the IBGP advertisements, you can attach the link-bandwidth community to communicate the bandwidth of the advertised external link. This is useful when multiple external links are available, and you want to do unequal load balancing over the links. You configure the link-bandwidth extended community on all ingress links of the AS. The bandwidth information in the link-bandwidth extended community is based on the configured bandwidth of the EBGP link. It is not based on the amount of traffic on the link. Junos OS supports BGP link-bandwidth and multipath load balancing, as described in Internet draft draft-ietf-idr-link-bandwidth-06, *BGP Link Bandwidth Extended Community*.

- [Requirements on page 384](#)
- [Overview on page 384](#)
- [Configuration on page 386](#)
- [Verification on page 390](#)

## Requirements

---

Before you begin:

- Configure the device interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure a routing policy that exports routes (such as direct routes or IGP routes) from the routing table into BGP.

## Overview

---

In this example, Device R1 is in AS 65000 and is connected to both Device R2 and Device R3, which are in AS 65001.

The example uses the bandwidth extended community.

By default, when BGP multipath is used, traffic is distributed equally among the several paths calculated. The bandwidth extended community allows an additional attribute to be added to BGP paths, thus allowing the traffic to be distributed unequally. The primary application is a scenario where multiple external paths exist for a given network with asymmetric bandwidth capabilities. In such a scenario, you can tag routes received with the bandwidth extended community. When BGP multipath (internal or external) operates among routes that contain the bandwidth attribute, the forwarding engine can unequally distribute traffic according to the bandwidth corresponding to each path.

When BGP has several candidate paths available for multipath purposes, BGP does not perform unequal cost load balancing according to the bandwidth community unless all candidate paths have this attribute.

The applicability of the bandwidth extended community is limited by the restrictions under which BGP multipath accepts multiple paths for consideration. Explicitly, the IGP distance, as far as BGP is concerned, between the router performing load balancing and the multiple exit points needs to be the same. This can be achieved by using a full mesh of label-switched paths (LSPs) that do not track the corresponding IGP metric. However, in a network in which the propagation delay of circuits is significant (for example, if long-haul circuits are present), it is often valuable to take into account the delay characteristics of different paths.

Configure the bandwidth community as follows:

```
[edit policy-options]  
user@host# set community members bandwidth:[1-65535]:[0-4294967295]
```

The first 16-bit number represents the local autonomous system. The second 32-bit number represents the link bandwidth in bytes per second.

For example:

```
[edit policy-options]  
user@host# show
```

```
community bw-t1 members bandwidth:10458:193000;
community bw-t3 members bandwidth:10458:5592000;
community bw-oc3 members bandwidth:10458:19440000;
```

Where 10458 is the local AS number. The values correspond to the bandwidth of the T1, T3, and OC-3 paths in bytes per second. The value specified as the bandwidth value does not need to correspond to the actual bandwidth of a specific interface. The balance factors used are calculated as a function of the total bandwidth specified. To tag a route with this extended community, define a policy statement, as follows:

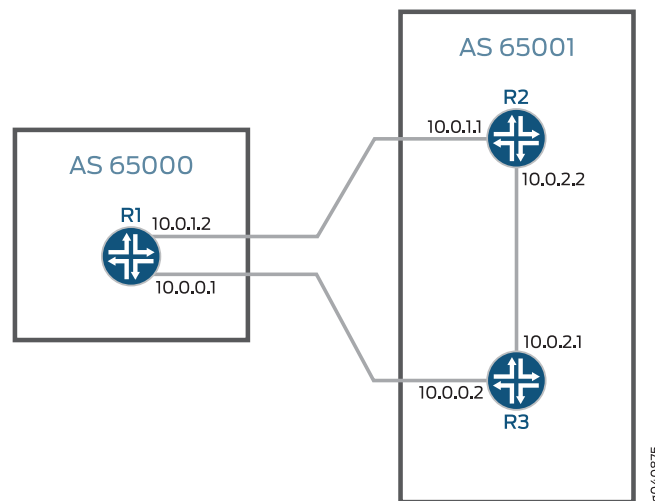
```
[edit policy-options]
user@host# show
policy-statement link-bw-t1 {
  then {
    community set bw-t1;
  }
  accept;
}
```

Apply this as an import policy on the BGP peering sessions facing the asymmetrical bandwidth links. Although in theory the community attribute can be added or removed at any point in the network, in the scenario described above, applying the community as an import policy in the EBGp peering session facing the external link allows for that attribute to influence the local multipath decision, and is potentially easier to manage.

### Topology

[Figure 41 on page 341](#) shows the topology used in this example.

**Figure 45: BGP Load Balancing**



[“CLI Quick Configuration” on page 341](#) shows the configuration for all of the devices in [Figure 41 on page 341](#). The section [“Step-by-Step Procedure” on page 342](#) 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 ge-1/2/0 unit 0 description R1->R3
set interfaces ge-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces ge-1/2/1 unit 0 description R1->R2
set interfaces ge-1/2/1 unit 0 family inet address 10.0.1.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group external type external
set protocols bgp group external import bw-dis
set protocols bgp group external peer-as 65001
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.1
set protocols bgp group external neighbor 10.0.0.2
set policy-options policy-statement bw-dis term a from protocol bgp
set policy-options policy-statement bw-dis term a from neighbor 10.0.1.1
set policy-options policy-statement bw-dis term a then community add bw-high
set policy-options policy-statement bw-dis term a then accept
set policy-options policy-statement bw-dis term b from protocol bgp
set policy-options policy-statement bw-dis term b from neighbor 10.0.0.2
set policy-options policy-statement bw-dis term b then community add bw-low
set policy-options policy-statement bw-dis term b then accept
set policy-options policy-statement loadbal from route-filter 10.0.0.0/16 orlonger
set policy-options policy-statement loadbal then load-balance per-packet
set policy-options community bw-high members bandwidth:65000:60000000
set policy-options community bw-low members bandwidth:65000:40000000
set routing-options autonomous-system 65000
set routing-options forwarding-table export loadbal

```

**Device R2**

```

set interfaces ge-1/2/0 unit 0 description R2->R1
set interfaces ge-1/2/0 unit 0 family inet address 10.0.1.1/30
set interfaces ge-1/2/1 unit 0 description R2->R3
set interfaces ge-1/2/1 unit 0 family inet address 10.0.2.2/30
set interfaces ge-1/2/1 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set interfaces lo0 unit 0 family iso address 49.0001.1921.6800.0002.00
set protocols bgp group external type external
set protocols bgp group external export bgp-default
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 65000
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.2
set protocols isis interface ge-1/2/1.0
set protocols isis interface lo0.0
set policy-options policy-statement bgp-default from protocol static
set policy-options policy-statement bgp-default from route-filter 172.16.0.0/16 exact
set policy-options policy-statement bgp-default then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options static route 172.16.0.0/16 discard
set routing-options static route 172.16.0.0/16 no-install

```

```
set routing-options autonomous-system 65001
```

**Device R3**

```

set interfaces ge-1/2/0 unit 0 description R3->R2
set interfaces ge-1/2/0 unit 0 family inet address 10.0.2.1/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces ge-1/2/1 unit 0 description R3->R1
set interfaces ge-1/2/1 unit 0 family inet address 10.0.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set interfaces lo0 unit 0 family iso address 49.0001.1921.6800.0003.00
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external export bgp-default
set protocols bgp group external peer-as 65000
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.0.1
set protocols isis interface ge-1/2/0.0
set protocols isis interface lo0.0
set policy-options policy-statement bgp-default from protocol static
set policy-options policy-statement bgp-default from route-filter 172.16.0.0/16 exact
set policy-options policy-statement bgp-default then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options static route 172.16.0.0/16 discard
set routing-options static route 172.16.0.0/16 no-install
set routing-options autonomous-system 65001

```

**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 BGP peer sessions:

1. Configure the interfaces.

```

user@R1# set ge-1/2/0 unit 0 description R1->R3
user@R1# set ge-1/2/0 unit 0 family inet address 10.0.0.1/30

```

```

user@R1# set ge-1/2/1 unit 0 description R1->R2
user@R1# set ge-1/2/1 unit 0 family inet address 10.0.1.2/30

```

```
user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure the BGP group.

```

[edit protocols bgp group external]
user@R1# set type external
user@R1# set import bw-dis
user@R1# set peer-as 65001
user@R1# set neighbor 10.0.1.1
user@R1# set neighbor 10.0.0.2

```

3. Enable the BGP group to use multiple paths.



**NOTE:** To disable the default check requiring that paths accepted by BGP multipath must have the same neighboring autonomous system (AS), include the `multiple-as` option. Use the `multiple-as` option if the neighbors are in different ASs.

```
[edit protocols bgp group external]
user@R1# set multipath
```

4. Configure the load-balancing policy.

```
[edit policy-options policy-statement loadbal]
user@R1# set from route-filter 10.0.0.0/16 orlonger
user@R1# set then load-balance per-packet
```

5. Apply the load-balancing policy.

```
[edit routing-options]
user@R1# set forwarding-table export loadbal
```

6. Configure the BGP community members.

This example assumes a bandwidth of 1 Gbps and allocates 60 percent to bw-high and 40 percent to bw-low. The reference bandwidth does not need to be the same as the link bandwidth.

```
[edit policy-options]
user@R1# set community bw-high members bandwidth:65000:60000000
user@R1# set community bw-low members bandwidth:65000:40000000
```

7. Configure the bandwidth distribution policy.

```
[edit policy-options bw-dis]
user@R1# set term a from protocol bgp
user@R1# set term a from neighbor 10.0.1.1
user@R1# set term a then community add bw-high
user@R1# set term a then accept
```

```
user@R1# set term b from protocol bgp
user@R1# set term b from neighbor 10.0.0.2
user@R1# set term b then community add bw-low
user@R1# set term b then accept
```

8. Configure the local autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

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

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

```
        description R1->R3;
        family inet {
            address 10.0.0.1/30;
        }
    }
}
ge-1/2/1 {
    unit 0 {
        description R1->R2;
        family inet {
            address 10.0.1.2/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.1/32;
        }
    }
}

user@R1# show protocols
bgp {
    group external {
        type external;
        import bw-dis;
        peer-as 65001;
        multipath;
        neighbor 10.0.1.1;
        neighbor 10.0.0.2;
    }
}

user@R1# show policy-options
policy-statement bw-dis {
    term a {
        from {
            protocol bgp;
            neighbor 10.0.1.1;
        }
        then {
            community add bw-high;
            accept;
        }
    }
    term b {
        from {
            protocol bgp;
            neighbor 10.0.0.2;
        }
        then {
            community add bw-low;
            accept;
        }
    }
}
```

```

policy-statement loadbal {
  from {
    route-filter 10.0.0.0/16 orlonger;
  }
  then {
    load-balance per-packet;
  }
}
community bw-high members bandwidth:65000:600000000;
community bw-low members bandwidth:65000:400000000;

user@R1# show routing-options
autonomous-system 65000;
forwarding-table {
  export loadbal;
}

```

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

## Verification

Confirm that the configuration is working properly:

### Verifying Routes

**Purpose** Verify that both routes are selected and that the next hops on the routes show a 60%/40% balance.

**Action** From operational mode, run the **show route protocol bgp detail** command.

```

user@R1> show route 172.16/16 protocol bgp detail
inet.0: 9 destinations, 13 routes (9 active, 0 holddown, 0 hidden)
172.16.0.0/16 (2 entries, 1 announced)
  *BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 262143
        Address: 0x93fc078
        Next-hop reference count: 3
        Source: 10.0.0.2
        Next hop: 10.0.0.2 via ge-1/2/0.0 balance 40%
        Next hop: 10.0.1.1 via ge-1/2/1.0 balance 60%, selected
        State: **Active Ext>
        Local AS: 65000 Peer AS: 65001
        Age: 3:22:55
        Task: BGP_65001.10.0.0.2+55344
        Announcement bits (1): 0-KRT
        AS path: 65001 I
        Communities: bandwidth:65000:400000000
        Accepted Multipath
        Localpref: 100
        Router ID: 192.168.0.3
  BGP   Preference: 170/-101
        Next hop type: Router, Next hop index: 658
        Address: 0x9260520
        Next-hop reference count: 4
        Source: 10.0.1.1
        Next hop: 10.0.1.1 via ge-1/2/1.0, selected
        State: <NotBest Ext>
        Inactive reason: Not Best in its group - Active preferred
        Local AS: 65000 Peer AS: 65001

```

```

Age: 3:22:55
Task: BGP_65001.10.0.1.1+62586
AS path: 65001 I
Communities: bandwidth:65000:60000000
Accepted MultipathContrib
Localpref: 100
Router ID: 192.168.0.2

user@R1> show route 10.0.2.0 protocol bgp detail
inet.0: 9 destinations, 13 routes (9 active, 0 holddown, 0 hidden)
10.0.2.0/30 (2 entries, 1 announced)
  *BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 262143
    Address: 0x93fc078
    Next-hop reference count: 3
    Source: 10.0.1.1
    Next hop: 10.0.0.2 via ge-1/2/0.0 balance 40%
    Next hop: 10.0.1.1 via ge-1/2/1.0 balance 60%, selected
    State: <Active Ext>
    Local AS: 65000 Peer AS: 65001
    Age: 3:36:37
    Task: BGP_65001.10.0.1.1+62586
    Announcement bits (1): 0-KRT
    AS path: 65001 I
    Communities: bandwidth:65000:60000000
    Accepted Multipath
    Localpref: 100
    Router ID: 192.168.0.2
  BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 657
    Address: 0x92604d8
    Next-hop reference count: 4
    Source: 10.0.0.2
    Next hop: 10.0.0.2 via ge-1/2/0.0, selected
    State: <NotBest Ext>
    Inactive reason: Not Best in its group - Active preferred
    Local AS: 65000 Peer AS: 65001
    Age: 3:36:36
    Task: BGP_65001.10.0.0.2+55344
    AS path: 65001 I
    Communities: bandwidth:65000:40000000
    Accepted MultipathContrib
    Localpref: 100
    Router ID: 192.168.0.3

```

**Meaning** The active path, denoted with an asterisk (\*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 172.16/16 destination.

Likewise, the active path, denoted with an asterisk (\*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 10.0.2.0 destination.

In both cases, the 10.0.1.1 next hop is copied from the inactive path to the active path.

The balance of 40 percent and 60 percent is shown in the **show route** output. This indicates that traffic is being distributed between two next hops and that 60 percent of the traffic is following the first path, while 40 percent is following the second path.

- Related Documentation**
- [Examples: Configuring BGP Multipath on page 365](#)

---

## Example: Advertising Multiple BGP Paths to a Destination

---

- [Understanding the Advertisement of Multiple Paths to a Single Destination in BGP on page 392](#)
- [Example: Advertising Multiple Paths in BGP on page 393](#)

### Understanding the Advertisement of Multiple Paths to a Single Destination in BGP

BGP peers advertise routes to each other in update messages. BGP stores its routes in the Junos OS routing table (**inet.0**). For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

Instead of advertising only the active path to a destination, you can configure BGP to advertise multiple paths to the destination. Within an autonomous system (AS), the availability of multiple exit points to reach a destination provides the following benefits:

- **Fault tolerance**—Path diversity leads to reduction in restoration time after failure. For instance, a border after receiving multiple paths to the same destination can precompute a backup path and have it ready so that when the primary path becomes invalid, the border routing device can use the backup to quickly restore connectivity. Without a backup path, the restoration time depends on BGP reconvergence, which includes withdraw and advertisement messages in the network before a new best path can be learned.
- **Load balancing**—The availability of multiple paths to reach the same destination enables load balancing of traffic, if the routing within the AS meets certain constraints.
- **Maintenance**—The availability of alternate exit points allows for graceful maintenance operation of routers.

The following limitations apply to advertising multiple routes in BGP:

- Address families supported:
  - IPv4 unicast (**family inet unicast**)
  - IPv6 unicast (**family inet6 unicast**)
  - IPv4 labeled unicast (**family inet labeled-unicast**)
  - IPv6 labeled unicast (**family inet6 labeled-unicast**)
- Internal BGP (IBGP) peers only. No support on external BGP (EBGP) peers.
- Master instance only. No support for routing instances.
- Graceful restart and nonstop active routing (NSR) are supported.
- No BGP Monitoring Protocol (BMP) support.

- No support for EBGP sessions between confederations.
- Prefix policies enable you to filter routes on a router that is configured to advertise multiple paths to a destination. Prefix policies can only match prefixes. They cannot match route attributes, and they cannot change the attributes of routes.

### Example: Advertising Multiple Paths in BGP

In this example, BGP routers are configured to advertise multiple paths instead of advertising only the active path. Advertising multiple paths in BGP is specified in Internet draft draft-ietf-idr-add-paths-04, *Advertisement of Multiple Paths in BGP*.

- [Requirements on page 393](#)
- [Overview on page 393](#)
- [Configuration on page 394](#)
- [Verification on page 413](#)

#### Requirements

This example uses the following hardware and software components:

- Eight BGP-enabled devices.
- Five of the BGP-enabled devices do not necessarily need to be routers. For example, they can be EX Series Ethernet Switches.
- Three of the BGP-enabled devices are configured to send multiple paths or receive multiple paths (or both send and receive multiple paths). These three BGP-enabled devices must be M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers.
- The three routers must be running Junos OS Release 11.4 or later.

#### Overview

The following statements are used for configuring multiple paths to a destination:

```
[edit protocols bgp group group-name family family]
add-path {
  receive;
  send {
    path-count number;
    prefix-policy [ policy-names ];
  }
}
```

In this example, Router R5, Router R6, and Router R7 redistribute static routes into BGP. Router R1 and Router R4 are route reflectors. Router R2 and Router R3 are clients to Route Reflector R1. Router R8 is a client to Route Reflector R4.

Route reflection is optional when multiple-path advertisement is enabled in BGP.

With the **add-path send path-count 6** configuration, Router R1 is configured to send up to six paths (per destination) to Router R4.

With the **add-path receive** configuration, Router R4 is configured to receive multiple paths from Router R1.

With the **add-path send path-count 6** configuration, Router R4 is configured to send up to six paths to Router R8.

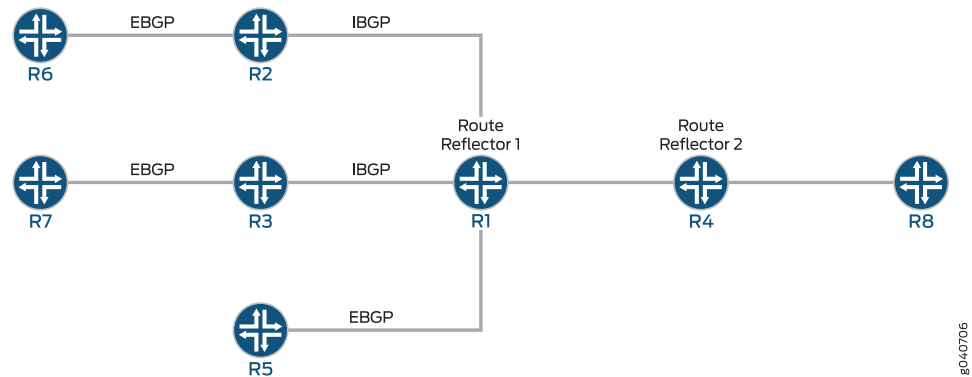
With the **add-path receive** configuration, Router R8 is configured to receive multiple paths from Router R4.

The **add-path send prefix-policy allow\_199** policy configuration (along with the corresponding route filter) limits Router R4 to sending multiple paths for only the 199.1.1.1/32 route.

### Topology Diagram

Figure 46 on page 394 shows the topology used in this example.

Figure 46: Advertisement of Multiple Paths in BGP



### Configuration

- [Configuring Router R1 on page 397](#)
- [Configuring Router R2 on page 400](#)
- [Configuring Router R3 on page 402](#)
- [Configuring Router R4 on page 404](#)
- [Configuring Router R5 on page 406](#)
- [Configuring Router R6 on page 408](#)
- [Configuring Router R7 on page 410](#)
- [Configuring Router R8 on page 411](#)
- [Results on page 412](#)

### 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.

**Router R1**

```

set interfaces fe-0/0/0 unit 12 family inet address 10.0.12.1/24
set interfaces fe-0/0/1 unit 13 family inet address 10.0.13.1/24
set interfaces fe-1/0/0 unit 14 family inet address 10.0.14.1/24

```

```

set interfaces fe-1/2/0 unit 15 family inet address 10.0.15.1/24
set interfaces lo0 unit 10 family inet address 10.0.0.10/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.10
set protocols bgp group rr cluster 10.0.0.10
set protocols bgp group rr neighbor 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.30
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.2 local-address 10.0.15.1
set protocols bgp group e1 neighbor 10.0.15.2 peer-as 2
set protocols bgp group rr_rr type internal
set protocols bgp group rr_rr local-address 10.0.0.10
set protocols bgp group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
  path-count 6
set protocols ospf area 0.0.0.0 interface lo0.10 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.12
set protocols ospf area 0.0.0.0 interface fe-0/0/1.13
set protocols ospf area 0.0.0.0 interface fe-1/0/0.14
set protocols ospf area 0.0.0.0 interface fe-1/2/0.15
set routing-options router-id 10.0.0.10
set routing-options autonomous-system 1

```

Router R2

```

set interfaces fe-1/2/0 unit 21 family inet address 10.0.12.2/24
set interfaces fe-1/2/1 unit 26 family inet address 10.0.26.1/24
set interfaces lo0 unit 20 family inet address 10.0.0.20/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.20 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.21
set protocols ospf area 0.0.0.0 interface fe-1/2/1.28
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1

```

Router R3

```

set interfaces fe-1/0/1 unit 31 family inet address 10.0.13.2/24
set interfaces fe-1/0/2 unit 37 family inet address 10.0.37.1/24
set interfaces lo0 unit 30 family inet address 10.0.0.30/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.30
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.30 passive
set protocols ospf area 0.0.0.0 interface fe-1/0/1.31
set protocols ospf area 0.0.0.0 interface fe-1/0/2.37
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1

```

Router R4

```

set interfaces fe-1/2/0 unit 41 family inet address 10.0.14.2/24
set interfaces fe-1/2/1 unit 48 family inet address 10.0.48.1/24
set interfaces lo0 unit 40 family inet address 10.0.0.40/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.40

```

```

set protocols bgp group rr family inet unicast add-path receive
set protocols bgp group rr neighbor 10.0.0.10
set protocols bgp group rr_client type internal
set protocols bgp group rr_client local-address 10.0.0.40
set protocols bgp group rr_client cluster 10.0.0.40
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  path-count 6
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  prefix-policy allow_199
set protocols ospf area 0.0.0.0 interface fe-1/2/0.41
set protocols ospf area 0.0.0.0 interface lo0.40 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/1.48
set routing-options autonomous-system 1
set policy-options policy-statement allow_199 from route-filter 199.1.1.1/32 exact
set policy-options policy-statement allow_199 term match_199 from prefix-list match_199
set policy-options policy-statement allow_199 then add-path send-count 20
set policy-options policy-statement allow_199 then accept

```

**Router R5**

```

set interfaces fe-1/2/0 unit 51 family inet address 10.0.15.2/24
set interfaces lo0 unit 50 family inet address 10.0.0.50/32
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.1 export s2b
set protocols bgp group e1 neighbor 10.0.15.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then as-path-expand 2
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject

```

**Router R6**

```

set interfaces fe-1/2/0 unit 62 family inet address 10.0.26.2/24
set interfaces lo0 unit 60 family inet address 10.0.0.60/32
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.1 export s2b
set protocols bgp group e1 neighbor 10.0.26.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject

```

**Router R7**

```

set interfaces fe-1/2/0 unit 73 family inet address 10.0.37.2/24
set interfaces lo0 unit 70 family inet address 10.0.0.70/32
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.1 export s2b
set protocols bgp group e1 neighbor 10.0.37.1 peer-as 1
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject

```

**Router R8**

```

set interfaces fe-1/2/0 unit 84 family inet address 10.0.48.2/24

```

```

set interfaces lo0 unit 80 family inet address 10.0.0.80/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.80
set protocols bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive
set protocols ospf area 0.0.0.0 interface lo0.80 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.84
set routing-options autonomous-system 1

```

### Configuring Router R1

**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 Router R1:

1. Configure the interfaces to Router R2, Router R3, Router R4, and Router R5, and configure the loopback (lo0) interface.

```

[edit interfaces]
user@R1# set fe-0/0/0 unit 12 family inet address 10.0.12.1/24

user@R1# set fe-0/0/1 unit 13 family inet address 10.0.13.1/24

user@R1# set fe-1/0/0 unit 14 family inet address 10.0.14.1/24

user@R1# set fe-1/2/0 unit 15 family inet address 10.0.15.1/24

user@R1# set lo0 unit 10 family inet address 10.0.0.10/32

```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

```

[edit protocols bgp]
user@R1# set group rr type internal
user@R1# set group rr local-address 10.0.0.10
user@R1# set group rr cluster 10.0.0.10
user@R1# set group rr neighbor 10.0.0.20
user@R1# set group rr neighbor 10.0.0.30

user@R1# set group rr_rr type internal
user@R1# set group rr_rr local-address 10.0.0.10

user@R1# set group e1 type external
user@R1# set group e1 neighbor 10.0.15.2 local-address 10.0.15.1
user@R1# set group e1 neighbor 10.0.15.2 peer-as 2

```

3. Configure Router R1 to send up to six paths to its neighbor, Router R4.

The destination of the paths can be any destination that Router R1 can reach through multiple paths.

```

[edit protocols bgp]
user@R1# set group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
path-count 6

```

4. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@R1# set area 0.0.0.0 interface lo0.10 passive
user@R1# set area 0.0.0.0 interface fe-0/0/0.12
user@R1# set area 0.0.0.0 interface fe-0/0/1.13
user@R1# set area 0.0.0.0 interface fe-1/0/0.14
user@R1# set area 0.0.0.0 interface fe-1/2/0.15
```

5. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@R1# set router-id 10.0.0.10
user@R1# set autonomous-system 1
```

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

```
user@R1# commit
```

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

```
user@R1# show interfaces
fe-0/0/0 {
  unit 12 {
    family inet {
      address 10.0.12.1/24;
    }
  }
}
fe-0/0/1 {
  unit 13 {
    family inet {
      address 10.0.13.1/24;
    }
  }
}
fe-1/0/0 {
  unit 14 {
    family inet {
      address 10.0.14.1/24;
    }
  }
}
fe-1/2/0 {
  unit 15 {
    family inet {
      address 10.0.15.1/24;
    }
  }
}
lo0 {
  unit 10 {
    family inet {
```

```
        address 10.0.0.10/32;
    }
}
}

user@R1# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.10;
    cluster 10.0.0.10;
    neighbor 10.0.0.20;
    neighbor 10.0.0.30;
  }
  group e1 {
    type external;
    neighbor 10.0.15.2 {
      local-address 10.0.15.1;
      peer-as 2;
    }
  }
  group rr_rr {
    type internal;
    local-address 10.0.0.10;
    neighbor 10.0.0.40 {
      family inet {
        unicast {
          add-path {
            send {
              path-count 6;
            }
          }
        }
      }
    }
  }
}
}

ospf {
  area 0.0.0.0 {
    interface lo0.10 {
      passive;
    }
    interface fe-0/0/0.12;
    interface fe-0/0/1.13;
    interface fe-1/0/0.14;
    interface fe-1/2/0.15;
  }
}

user@R1# show routing-options
router-id 10.0.0.10;
autonomous-system 1;
```

### *Configuring Router R2*

#### **Step-by-Step Procedure**

To configure Router R2:

1. Configure the loopback (lo0) interface and the interfaces to Router R6 and Router R1.

```
[edit interfaces]
```

```
user@R2# set fe-1/2/0 unit 21 family inet address 10.0.12.2/24
```

```
user@R2# set fe-1/2/1 unit 26 family inet address 10.0.26.1/24
```

```
user@R2# set lo0 unit 20 family inet address 10.0.0.20/32
```

2. Configure BGP and OSPF on Router R2's interfaces.

```
[edit protocols]
```

```
user@R2# set bgp group rr type internal
```

```
user@R2# set bgp group rr local-address 10.0.0.20
```

```
user@R2# set bgp group e1 type external
```

```
user@R2# set bgp group e1 neighbor 10.0.26.2 peer-as 2
```

```
user@R2# set ospf area 0.0.0.0 interface lo0.20 passive
```

```
user@R2# set ospf area 0.0.0.0 interface fe-1/2/0.21
```

```
user@R2# set ospf area 0.0.0.0 interface fe-1/2/1.28
```

3. For routes sent from Router R2 to Router R1, advertise Router R2 as the next hop, because Router R1 does not have a route to Router R6's address on the 10.0.26.0/24 network.

```
[edit]
```

```
user@R2# set policy-options policy-statement set_nh_self then next-hop self
```

```
user@R2# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
```

```
user@R2# set routing-options autonomous-system 1
```

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

```
user@R2# commit
```

#### **Results**

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 21 {
    family inet {
      address 10.0.12.2/24;
```

```

    }
  }
}
fe-1/2/1 {
  unit 26 {
    family inet {
      address 10.0.26.1/24;
    }
  }
}
lo0 {
  unit 20 {
    family inet {
      address 10.0.0.20/32;
    }
  }
}
}

user@R2# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.20;
    neighbor 10.0.0.10 {
      export set_nh_self;
    }
  }
  group e1 {
    type external;
    neighbor 10.0.26.2 {
      peer-as 2;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.20 {
      passive;
    }
    interface fe-1/2/0.21;
    interface fe-1/2/1.28;
  }
}

user@R2# show policy-options
policy-statement set_nh_self {
  then {
    next-hop self;
  }
}

user@R2# show routing-options
autonomous-system 1;

```

### *Configuring Router R3*

#### **Step-by-Step Procedure**

To configure Router R3:

1. Configure the loopback (lo0) interface and the interfaces to Router R7 and Router R1.

```
[edit interfaces]
```

```
user@R3# set fe-1/0/1 unit 31 family inet address 10.0.13.2/24
```

```
user@R3# set fe-1/0/2 unit 37 family inet address 10.0.37.1/24
```

```
user@R3# set lo0 unit 30 family inet address 10.0.0.30/32
```

2. Configure BGP and OSPF on Router R3's interfaces.

```
[edit protocols]
```

```
user@R3# set bgp group rr type internal
```

```
user@R3# set bgp group rr local-address 10.0.0.30
```

```
user@R3# set bgp group e1 type external
```

```
user@R3# set bgp group e1 neighbor 10.0.37.2 peer-as 2
```

```
user@R3# set ospf area 0.0.0.0 interface lo0.30 passive
```

```
user@R3# set ospf area 0.0.0.0 interface fe-1/0/1.31
```

```
user@R3# set ospf area 0.0.0.0 interface fe-1/0/2.37
```

3. For routes sent from Router R3 to Router R1, advertise Router R3 as the next hop, because Router R1 does not have a route to Router R7's address on the 10.0.37.0/24 network.

```
[edit]
```

```
user@R3# set policy-options policy-statement set_nh_self then next-hop self
```

```
user@R3# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
```

```
user@R3# set routing-options autonomous-system 1
```

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

```
user@R3# commit
```

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

```
user@R3# show interfaces
fe-1/0/1 {
  unit 31 {
    family inet {
      address 10.0.13.2/24;
```

```
    }
  }
}
fe-1/0/2 {
  unit 37 {
    family inet {
      address 10.0.37.1/24;
    }
  }
}
lo0 {
  unit 30 {
    family inet {
      address 10.0.0.30/32;
    }
  }
}

user@R3# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.30;
    neighbor 10.0.0.10 {
      export set_nh_self;
    }
  }
  group e1 {
    type external;
    neighbor 10.0.37.2 {
      peer-as 2;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.30 {
      passive;
    }
    interface fe-1/0/1.31;
    interface fe-1/0/2.37;
  }
}

user@R3# show policy-options
policy-statement set_nh_self {
  then {
    next-hop self;
  }
}

user@R3# show routing-options
autonomous-system 1;
```

**Configuring Router R4****Step-by-Step Procedure**

To configure Router R4:

1. Configure the interfaces to Router R1 and Router R8, and configure the loopback (lo0) interface.

```
[edit interfaces]
```

```
user@R4# set fe-1/2/0 unit 41 family inet address 10.0.14.2/24
```

```
user@R4# set fe-1/2/1 unit 48 family inet address 10.0.48.1/24
```

```
user@R4# set lo0 unit 40 family inet address 10.0.0.40/32
```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

```
[edit protocols bgp]
```

```
user@R4# set group rr type internal
```

```
user@R4# set group rr local-address 10.0.0.40
```

```
user@R4# set group rr neighbor 10.0.0.10
```

```
user@R4# set group rr_client type internal
```

```
user@R4# set group rr_client local-address 10.0.0.40
```

```
user@R4# set group rr_client cluster 10.0.0.40
```

3. Configure Router R4 to send up to six paths to its neighbor, Router R8.

The destination of the paths can be any destination that Router R4 can reach through multiple paths.

```
[edit protocols bgp]
```

```
user@R4# set group rr_client neighbor 10.0.0.80 family inet unicast add-path send path-count 6
```

4. Configure Router R4 to receive multiple paths from its neighbor, Router R1.

The destination of the paths can be any destination that Router R1 can reach through multiple paths.

```
[edit protocols bgp group rr family inet unicast]
```

```
user@R4# set add-path receive
```

5. Configure OSPF on the interfaces.

```
[edit protocols ospf area 0.0.0.0]
```

```
user@R4# set interface fe-1/2/0.41
```

```
user@R4# set interface lo0.40 passive
```

```
user@R4# set interface fe-1/2/1.48
```

6. Configure a policy that allows Router R4 to send Router R8 multiple paths to the 199.1.1.1/32 route.

- Router R4 receives multiple paths for the 198.1.1.1/32 route and the 199.1.1.1/32 route. However, because of this policy, Router R4 only sends multiple paths for the 199.1.1.1/32 route.

```
[edit protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast]
```

```
user@R4# set add-path send prefix-policy allow_199
```

```
[edit policy-options policy-statement allow_199]
user@R4# set from route-filter 199.1.1/32 exact
user@R4# set then accept
```

- Router R4 can also be configured to send up-to 20 BGP **add-path** routes for a subset of *add-path advertised prefixes*.

```
[edit policy-options policy-statement allow_199]
user@R4# set term match_199 from prefix-list match_199
user@R4# set then add-path send-count 20
```

7. Configure the autonomous system number.

```
[edit routing-options]
user@R4# set autonomous-system 1
```

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

```
user@R4# commit
```

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

```
user@R4# show interfaces
fe-1/2/0 {
  unit 41 {
    family inet {
      address 10.0.14.2/24;
    }
  }
}
fe-1/2/1 {
  unit 48 {
    family inet {
      address 10.0.48.1/24;
    }
  }
}
lo0 {
  unit 40 {
    family inet {
      address 10.0.0.40/32;
    }
  }
}
```

```
user@R4# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.40;
    family inet {
      unicast {
        add-path {
          receive;
        }
      }
    }
  }
}
```

```

    }
  }
}
neighbor 10.0.0.10;
}
group rr_client {
  type internal;
  local-address 10.0.0.40;
  cluster 10.0.0.40;
  neighbor 10.0.0.80 {
    family inet {
      unicast {
        add-path {
          send {
            path-count 6;
            prefix-policy allow_199;
          }
        }
      }
    }
  }
}
}
}
}
}
}
ospf {
  area 0.0.0.0 {
    interface lo0.40 {
      passive;
    }
    interface fe-1/2/0.41;
    interface fe-1/2/1.48;
  }
}
}
}
user@R4# show policy-options
policy-statement allow_199 {
  from {
    route-filter 199.1.1.1/32 exact;
  }
  from term match_199 {
    prefix-list match_199;
  }
  then add-path send-count 20;
  then accept;
}
}
user@R4# show routing-options
autonomous-system 1;
```

### Configuring Router R5

## Step-by-Step Procedure

To configure Router R5:

1. Configure the loopback (lo0) interface and the interface to Router R1.

[edit interfaces]

```
user@R5# set fe-1/2/0 unit 51 family inet address 10.0.15.2/24
```

```
user@R5# set lo0 unit 50 family inet address 10.0.0.50/32
```

2. Configure BGP on Router R5's interface.

```
[edit protocols bgp group e1]
user@R5# set type external
user@R5# set neighbor 10.0.15.1 peer-as 1
```

3. Create static routes for redistribution into BGP.

```
[edit routing-options]
user@R5# set static route 199.1.1.1/32 reject
user@R5# set static route 198.1.1.1/32 reject
```

4. Redistribute static and direct routes into BGP.

```
[edit protocols bgp group e1 neighbor 10.0.15.1]
user@R5# set export s2b
```

```
[edit policy-options policy-statement s2b]
user@R5# set from protocol static
user@R5# set from protocol direct
user@R5# set then as-path-expand 2
user@R5# set then accept
```

5. Configure the autonomous system number.

```
[edit routing-options]
user@R5# set autonomous-system 2
```

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

```
user@R5# commit
```

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

```
user@R5# show interfaces
fe-1/2/0 {
  unit 51 {
    family inet {
      address 10.0.15.2/24;
    }
  }
}
lo0 {
  unit 50 {
    family inet {
      address 10.0.0.50/32;
    }
  }
}

user@R5# show protocols
bgp {
  group e1 {
```

```

        type external;
        neighbor 10.0.15.1 {
            export s2b;
            peer-as 1;
        }
    }
}

user@R5# show policy-options
policy-statement s2b {
    from protocol [ static direct ];
    then {
        as-path-expand 2;
        accept;
    }
}

user@R5# show routing-options
static {
    route 198.1.1.1/32 reject;
    route 199.1.1.1/32 reject;
}
autonomous-system 2;

```

### Configuring Router R6

#### Step-by-Step Procedure

To configure Router R6:

1. Configure the loopback (lo0) interface and the interface to Router R2.
 

```

[edit interfaces]
user@R6# set fe-1/2/0 unit 62 family inet address 10.0.26.2/24

user@R6# set lo0 unit 60 family inet address 10.0.0.60/32
            
```
2. Configure BGP on Router R6's interface.
 

```

[edit protocols]
user@R6# set bgp group e1 type external
user@R6# set bgp group e1 neighbor 10.0.26.1 peer-as 1
            
```
3. Create static routes for redistribution into BGP.
 

```

[edit]
user@R6# set routing-options static route 199.1.1.1/32 reject
user@R6# set routing-options static route 198.1.1.1/32 reject
            
```
4. Redistribute static and direct routes from Router R6's routing table into BGP.
 

```

[edit protocols bgp group e1 neighbor 10.0.26.1]
user@R6# set export s2b

[edit policy-options policy-statement s2b]
user@R6# set from protocol static
user@R6# set from protocol direct
user@R6# set then accept
            
```

5. Configure the autonomous system number.

```
[edit routing-options]
user@R6# set autonomous-system 2
```

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

```
user@R6# commit
```

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

```
user@R6# show interfaces
fe-1/2/0 {
  unit 62 {
    family inet {
      address 10.0.26.2/24;
    }
  }
}
lo0 {
  unit 60 {
    family inet {
      address 10.0.0.60/32;
    }
  }
}

user@R6# show protocols
bgp {
  group e1 {
    type external;
    neighbor 10.0.26.1 {
      export s2b;
      peer-as 1;
    }
  }
}

user@R6# show policy-options
policy-statement s2b {
  from protocol [ static direct ];
  then accept;
}

user@R6# show routing-options
static {
  route 198.1.1.1/32 reject;
  route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

### *Configuring Router R7*

#### **Step-by-Step Procedure**

To configure Router R7:

1. Configure the loopback (lo0) interface and the interface to Router R3.  
  
[edit interfaces]  
user@R7# set fe-1/2/0 unit 73 family inet address 10.0.37.2/24  
  
user@R7# set lo0 unit 70 family inet address 10.0.0.70/32
2. Configure BGP on Router R7's interface.  
  
[edit protocols bgp group e1]  
user@R7# set type external  
user@R7# set neighbor 10.0.37.1 peer-as 1
3. Create a static route for redistribution into BGP.  
  
[edit]  
user@R7# set routing-options static route 199.1.1.1/32 reject
4. Redistribute static and direct routes from Router R7's routing table into BGP.  
  
[edit protocols bgp group e1 neighbor 10.0.37.1]  
user@R7# set export s2b  
  
[edit policy-options policy-statement s2b]  
user@R7# set from protocol static  
user@R7# set from protocol direct  
user@R7# set then accept
5. Configure the autonomous system number.  
  
[edit routing-options]  
user@R7# set autonomous-system 2
6. If you are done configuring the device, commit the configuration.  
  
user@R7# commit

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

```
user@R7# show interfaces
fe-1/2/0 {
  unit 73 {
    family inet {
      address 10.0.37.2/24;
    }
  }
}
lo0 {
  unit 70 {
    family inet {
```

```

        address 10.0.0.70/32;
    }
}
}

user@R7# show protocols
bgp {
    group e1 {
        type external;
        neighbor 10.0.37.1 {
            export s2b;
            peer-as 1;
        }
    }
}

user@R7# show policy-options
policy-statement s2b {
    from protocol [ static direct ];
    then accept;
}

user@R7# show routing-options
static {
    route 199.1.1.1/32 reject;
}
autonomous-system 2;

```

### Configuring Router R8

#### Step-by-Step Procedure

To configure Router R8:

1. Configure the loopback (lo0) interface and the interface to Router R4.

```

[edit interfaces]
user@R8# set fe-1/2/0 unit 84 family inet address 10.0.48.2/24

user@R8# set lo0 unit 80 family inet address 10.0.0.80/32

```

2. Configure BGP and OSPF on Router R8's interface.

```

[edit protocols]
user@R8# set bgp group rr type internal
user@R8# set bgp group rr local-address 10.0.0.80

user@R8# set ospf area 0.0.0.0 interface lo0.80 passive
user@R8# set ospf area 0.0.0.0 interface fe-1/2/0.84

```

3. Configure Router R8 to receive multiple paths from its neighbor, Router R4.

The destination of the paths can be any destination that Router R4 can reach through multiple paths.

```

[edit protocols]
user@R8# set bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive

```

4. Configure the autonomous system number.

```

[edit]

```

```
user@R8# set routing-options autonomous-system 1
```

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

```
user@R8# commit
```

### Results

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

```
user@R8# show interfaces
fe-1/2/0 {
  unit 84 {
    family inet {
      address 10.0.48.2/24;
    }
  }
}
lo0 {
  unit 80 {
    family inet {
      address 10.0.0.80/32;
    }
  }
}

user@R8# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.80;
    neighbor 10.0.0.40 {
      family inet {
        unicast {
          add-path {
            receive;
          }
        }
      }
    }
  }
}

ospf {
  area 0.0.0.0 {
    interface lo0.80 {
      passive;
    }
    interface fe-1/2/0.84;
  }
}

user@R8# show routing-options
autonomous-system 1;
```

## Verification

Confirm that the configuration is working properly.

- [Verifying That the BGP Peers Have the Ability to Send and Receive Multiple Paths on page 413](#)
- [Verifying That Router R1 Is Advertising Multiple Paths on page 413](#)
- [Verifying That Router R4 Is Receiving and Advertising Multiple Paths on page 414](#)
- [Verifying That Router R8 Is Receiving Multiple Paths on page 415](#)
- [Checking the Path ID on page 415](#)

### *Verifying That the BGP Peers Have the Ability to Send and Receive Multiple Paths*

**Purpose** Make sure that one or both of the following strings appear in the output of the `show bgp neighbor` command:

- NLRI's for which peer can receive multiple paths: inet-unicast
- NLRI's for which peer can send multiple paths: inet-unicast

**Action**

```

user@R1> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1      Local: 10.0.0.10+65237 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...  NLRI's for which peer can receive multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.10
Peer: 10.0.0.10+65237 AS 1    Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...
  NLRI's for which peer can send multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.80
Peer: 10.0.0.80+55416 AS 1    Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established (route reflector client)Flags: <Sync>
...
  NLRI's for which peer can receive multiple paths: inet-unicast
...

user@R8> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1      Local: 10.0.0.80+55416 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...
  NLRI's for which peer can send multiple paths: inet-unicast
...

```

### *Verifying That Router R1 Is Advertising Multiple Paths*

**Purpose** Make sure that multiple paths to the 198.1.1/32 destination and multiple paths to the 199.1.1/32 destination are advertised to Router R4.

**Action** user@R1> show route advertising-protocol bgp 10.0.0.40  
 inet.0: 21 destinations, 25 routes (21 active, 0 holddown, 0 hidden)

Prefix	Nexthop	MED	Lc1pref	AS path
* 10.0.0.50/32	10.0.15.2		100	2 2 I
* 10.0.0.60/32	10.0.0.20		100	2 I
* 10.0.0.70/32	10.0.0.30		100	2 I
* 198.1.1.1/32	10.0.0.20		100	2 I
	10.0.15.2		100	2 2 I
* 199.1.1.1/32	10.0.0.20		100	2 I
	10.0.0.30		100	2 I
	10.0.15.2		100	2 2 I
* 200.1.1.0/30	10.0.0.20		100	2 I

**Meaning** When you see one prefix and more than one next hop, it means that multiple paths are advertised to Router R4.

#### *Verifying That Router R4 Is Receiving and Advertising Multiple Paths*

**Purpose** Make sure that multiple paths to the 199.1.1.1/32 destination are received from Router R1 and advertised to Router R8. Make sure that multiple paths to the 198.1.1.1/32 destination are received from Router R1, but only one path to this destination is advertised to Router R8.

**Action** user@R4> show route receive-protocol bgp 10.0.0.10  
 inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)

Prefix	Nexthop	MED	Lc1pref	AS path
* 10.0.0.50/32	10.0.15.2		100	2 2 I
* 10.0.0.60/32	10.0.0.20		100	2 I
* 10.0.0.70/32	10.0.0.30		100	2 I
* 198.1.1.1/32	10.0.0.20		100	2 I
	10.0.15.2		100	2 2 I
* 199.1.1.1/32	10.0.0.20		100	2 I
	10.0.0.30		100	2 I
	10.0.15.2		100	2 2 I
* 200.1.1.0/30	10.0.0.20		100	2 I

user@R4> show route advertising-protocol bgp 10.0.0.80  
 inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)

Prefix	Nexthop	MED	Lc1pref	AS path
* 10.0.0.50/32	10.0.15.2		100	2 2 I
* 10.0.0.60/32	10.0.0.20		100	2 I
* 10.0.0.70/32	10.0.0.30		100	2 I
* 198.1.1.1/32	10.0.0.20		100	2 I
* 199.1.1.1/32	10.0.0.20		100	2 I
	10.0.0.30		100	2 I
	10.0.15.2		100	2 2 I
* 200.1.1.0/30	10.0.0.20		100	2 I

**Meaning** The **show route receive-protocol** command shows that Router R4 receives two paths to the 198.1.1.1/32 destination and three paths to the 199.1.1.1/32 destination. The **show route advertising-protocol** command shows that Router R4 advertises only one path to the 198.1.1.1/32 destination and advertises all three paths to the 199.1.1.1/32 destination.

Because of the prefix policy that is applied to Router R4, Router R4 does not advertise multiple paths to the 198.1.1.1/32 destination. Router R4 advertises only one path to the 198.1.1.1/32 destination even though it receives multiple paths to this destination.

#### *Verifying That Router R8 Is Receiving Multiple Paths*

**Purpose** Make sure that Router R8 receives multiple paths to the 199.1.1.1/32 destination through Router R4. Make sure that Router R8 receives only one path to the 198.1.1.1/32 destination through Router R4.

**Action** user@R8> show route receive-protocol bgp 10.0.0.40  
 inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)

Prefix	Nexthop	MED	Lc1pref	AS path
* 10.0.0.50/32	10.0.15.2		100	2 2 I
* 10.0.0.60/32	10.0.0.20		100	2 I
* 10.0.0.70/32	10.0.0.30		100	2 I
* 198.1.1.1/32	10.0.0.20		100	2 I
* 199.1.1.1/32	10.0.0.20		100	2 I
	10.0.0.30		100	2 I
	10.0.15.2		100	2 2 I
* 200.1.1.0/30	10.0.0.20		100	2 I

#### *Checking the Path ID*

**Purpose** On the downstream devices, Router R4 and Router R8, verify that a path ID uniquely identifies the path. Look for the **Addpath Path ID:** string.

**Action** user@R4> show route 199.1.1.1/32 detail

```
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 3 announced)
  *BGP
    Preference: 170/-101
    Next hop type: Indirect
    Next-hop reference count: 9
    Source: 10.0.0.10
    Next hop type: Router, Next hop index: 676
    Next hop: 10.0.14.1 via lt-1/2/0.41, selected
    Protocol next hop: 10.0.0.20
    Indirect next hop: 92041c8 262146
    State: <Active Int Ext>
    Local AS: 1 Peer AS: 1
    Age: 1:44:37 Metric2: 2
    Task: BGP_1.10.0.0.10+65237
    Announcement bits (3): 2-KRT 3-BGP RT Background 4-Resolve tree

  1
    AS path: 2 I (Originator) Cluster list: 10.0.0.10
    AS path: Originator ID: 10.0.0.20
    Accepted
    Localpref: 100
    Router ID: 10.0.0.10
    Addpath Path ID: 1
  *BGP
    Preference: 170/-101
    Next hop type: Indirect
    Next-hop reference count: 4
    Source: 10.0.0.10
    Next hop type: Router, Next hop index: 676
    Next hop: 10.0.14.1 via lt-1/2/0.41, selected
    Protocol next hop: 10.0.0.30
    Indirect next hop: 92042ac 262151
    State: <NotBest Int Ext>
    Inactive reason: Not Best in its group - Router ID
    Local AS: 1 Peer AS: 1
    Age: 1:44:37 Metric2: 2
    Task: BGP_1.10.0.0.10+65237
    Announcement bits (1): 3-BGP RT Background
    AS path: 2 I (Originator) Cluster list: 10.0.0.10
    AS path: Originator ID: 10.0.0.30
    Accepted
    Localpref: 100
    Router ID: 10.0.0.10
    Addpath Path ID: 2
  *BGP
    Preference: 170/-101
    Next hop type: Indirect
    Next-hop reference count: 4
    Source: 10.0.0.10
    Next hop type: Router, Next hop index: 676
    Next hop: 10.0.14.1 via lt-1/2/0.41, selected
    Protocol next hop: 10.0.15.2
    Indirect next hop: 92040e4 262150
    State: <Int Ext>
    Inactive reason: AS path
    Local AS: 1 Peer AS: 1
    Age: 1:44:37 Metric2: 2
    Task: BGP_1.10.0.0.10+65237
    Announcement bits (1): 3-BGP RT Background
    AS path: 2 2 I
    Accepted
```

```

Localpref: 100
Router ID: 10.0.0.10
Addpath Path ID: 3

```

```
user@R8> show route 199.1.1.1/32 detail
```

```

inet.0: 17 destinations, 19 routes (17 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 1 announced)
*BGP   Preference: 170/-101
      Next hop type: Indirect
      Next-hop reference count: 9
      Source: 10.0.0.40
      Next hop type: Router, Next hop index: 1045
      Next hop: 10.0.48.1 via lt-1/2/0.84, selected
      Protocol next hop: 10.0.0.20
      Indirect next hop: 91fc0e4 262148
      State: <Active Int Ext>
      Local AS:      1 Peer AS:      1
      Age: 1:56:51   Metric2: 3
      Task: BGP_1.10.0.0.40+179
      Announcement bits (2): 2-KRT 4-Resolve tree 1
      AS path: 2 I (Originator) Cluster list: 10.0.0.40 10.0.0.10
      AS path: Originator ID: 10.0.0.20
      Accepted
      Localpref: 100
      Router ID: 10.0.0.40
      Addpath Path ID: 1
BGP   Preference: 170/-101
      Next hop type: Indirect
      Next-hop reference count: 4
      Source: 10.0.0.40
      Next hop type: Router, Next hop index: 1045
      Next hop: 10.0.48.1 via lt-1/2/0.84, selected
      Protocol next hop: 10.0.0.30
      Indirect next hop: 91fc1c8 262152
      State: <NotBest Int Ext>
      Inactive reason: Not Best in its group - Router ID
      Local AS:      1 Peer AS:      1
      Age: 1:56:51   Metric2: 3
      Task: BGP_1.10.0.0.40+179
      AS path: 2 I (Originator) Cluster list: 10.0.0.40 10.0.0.10
      AS path: Originator ID: 10.0.0.30
      Accepted
      Localpref: 100
      Router ID: 10.0.0.40
      Addpath Path ID: 2
BGP   Preference: 170/-101
      Next hop type: Indirect
      Next-hop reference count: 4
      Source: 10.0.0.40
      Next hop type: Router, Next hop index: 1045
      Next hop: 10.0.48.1 via lt-1/2/0.84, selected
      Protocol next hop: 10.0.15.2
      Indirect next hop: 91fc2ac 262153
      State: <Int Ext>
      Inactive reason: AS path
      Local AS:      1 Peer AS:      1
      Age: 1:56:51   Metric2: 3
      Task: BGP_1.10.0.0.40+179
      AS path: 2 2 I (Originator) Cluster list: 10.0.0.40
      AS path: Originator ID: 10.0.0.10

```

Accepted  
Localpref: 100  
Router ID: 10.0.0.40  
Addpath Path ID: 3

- Related Documentation**
- [Understanding External BGP Peering Sessions on page 33](#)
  - [BGP Configuration Overview on page 12](#)

## CHAPTER 9

# IBGP Scaling Configuration

- [Example: Configuring BGP Route Reflectors on page 419](#)
- [Example: Configuring a Route Reflector that Belongs to Two Different Clusters on page 436](#)
- [Example: Configuring BGP Confederations on page 441](#)

### Example: Configuring BGP Route Reflectors

---

- [Understanding BGP Route Reflectors on page 419](#)
- [Example: Configuring a Route Reflector on page 421](#)

### Understanding BGP Route Reflectors

Because of the internal BGP (IBGP) full-mesh requirement, most networks use route reflectors to simplify configuration. The formula to compute the number of sessions required for a full mesh is  $v * (v - 1) / 2$ , where  $v$  is the number of BGP-enabled devices. The full-mesh model does not scale well. Using a route reflector, you group routers into clusters, which are identified by numeric identifiers unique to the autonomous system (AS). Within the cluster, you must configure a BGP session from a single router (the route reflector) to each internal peer. With this configuration, the IBGP full-mesh requirement is met.

To use route reflection in an AS, you designate one or more routers as a route reflector—typically, one per point of presence (POP). Route reflectors have the special BGP ability to readvertise routes learned from an internal peer to other internal peers. So rather than requiring all internal peers to be fully meshed with each other, route reflection requires only that the route reflector be fully meshed with all internal peers. The route reflector and all of its internal peers form a cluster, as shown in [Figure 47 on page 420](#).



**NOTE:** For some Juniper Networks devices, you must have an Advanced BGP Feature license installed on each device that uses a route reflector. For license details, see the *Junos OS Initial Configuration Guide for Security Devices*.

Figure 47: Simple Route Reflector Topology (One Cluster)

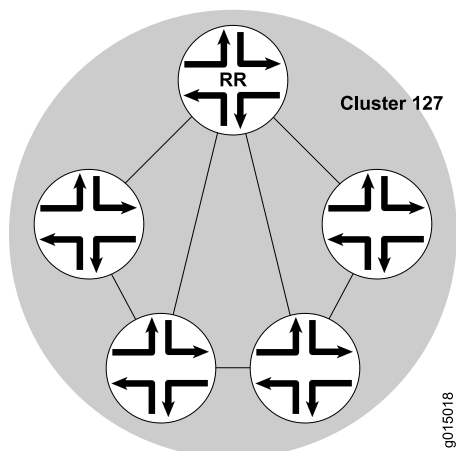


Figure 47 on page 420 shows Router RR configured as the route reflector for Cluster 127. The other routers are designated internal peers within the cluster. BGP routes are advertised to Router RR by any of the internal peers. RR then readvertises those routes to all other peers within the cluster.

You can configure multiple clusters and link them by configuring a full mesh of route reflectors (see Figure 48 on page 420).

Figure 48: Basic Route Reflection (Multiple Clusters)

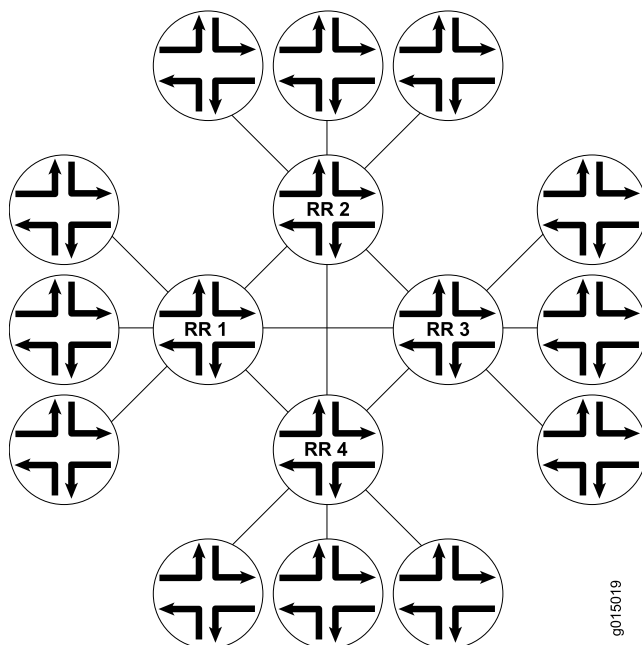
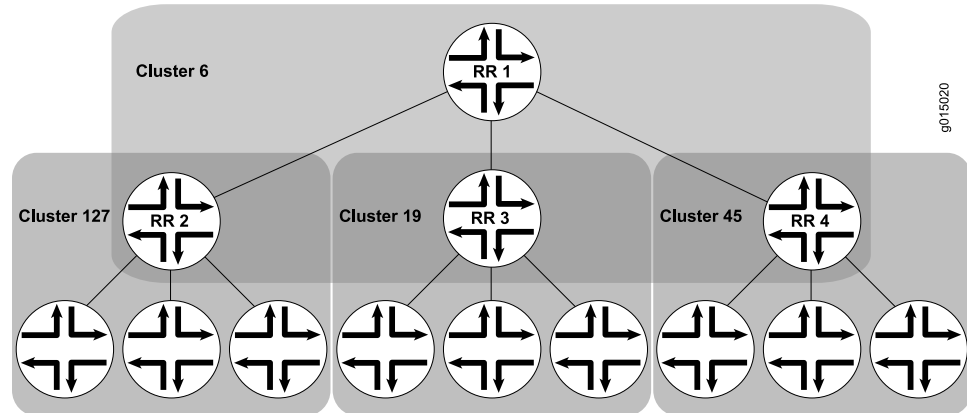


Figure 48 on page 420 shows Route Reflectors RR 1, RR 2, RR 3, and RR 4 as fully meshed internal peers. When a router advertises a route to RR 1, RR 1 readvertises the route to the other route reflectors, which, in turn, readvertise the route to the remaining routers within the AS. Route reflection allows the route to be propagated throughout the AS without the scaling problems created by the full mesh requirement.

However, as clusters become large, a full mesh with a route reflector becomes difficult to scale, as does a full mesh between route reflectors. To help offset this problem, you can group clusters of routers together into clusters of clusters for hierarchical route reflection (see [Figure 49 on page 421](#)).

**Figure 49: Hierarchical Route Reflection (Clusters of Clusters)**



[Figure 49 on page 421](#) shows RR 2, RR 3, and RR 4 as the route reflectors for Clusters 127, 19, and 45, respectively. Rather than fully mesh those route reflectors, the network administrator has configured them as part of another cluster (Cluster 6) for which RR 1 is the route reflector. When a router advertises a route to RR 2, RR 2 readvertises the route to all the routers within its own cluster, and then readvertises the route to RR 1. RR 1 readvertises the route to the routers in its cluster, and those routers propagate the route down through their clusters.

### Example: Configuring a Route Reflector

This example shows how to configure a route reflector.

- [Requirements on page 421](#)
- [Overview on page 421](#)
- [Configuration on page 423](#)
- [Verification on page 431](#)

#### Requirements

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

#### Overview

Generally, internal BGP (IBGP)-enabled devices need to be fully meshed, because IBGP does not readvertise updates to other IBGP-enabled devices. The full mesh is a logical mesh achieved through configuration of multiple **neighbor** statements on each IBGP-enabled device. The full mesh is not necessarily a physical full mesh. Maintaining a full mesh (logical or physical) does not scale well in large deployments.

Figure 50 on page 423 shows an IBGP network with Device A acting as a route reflector. Device B and Device C are clients of the route reflector. Device D and Device E are outside the cluster, so they are nonclients of the route reflector.

On Device A (the route reflector), you must form peer relationships with all of the IBGP-enabled devices by including the **neighbor** statement for the clients (Device B and Device C) and the nonclients (Device D and Device E). You must also include the **cluster** statement and a cluster identifier. The cluster identifier can be any 32-bit value. This example uses the loopback interface IP address of the route reflector.

On Device B and Device C, the route reflector clients, you only need one **neighbor** statement that forms a peer relationship with the route reflector, Device A.

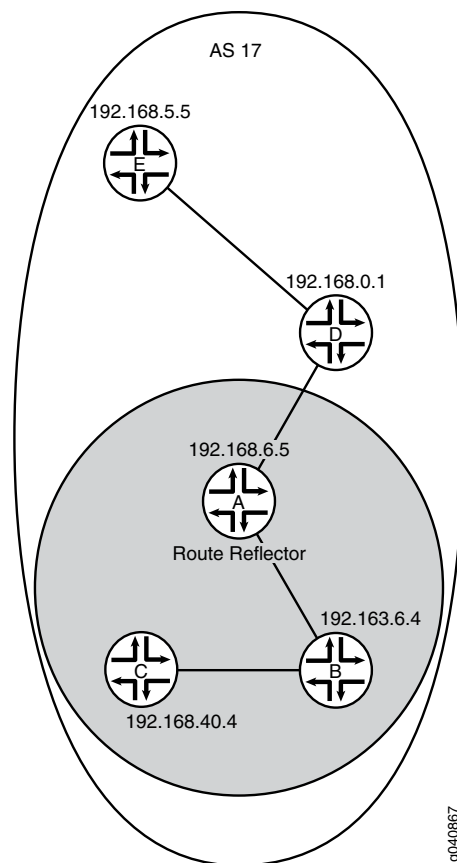
On Device D and Device E, the nonclients, you need a **neighbor** statement for each nonclient device (D-to-E and E-to-D). You also need a **neighbor** statement for the route reflector (D-to-A and E-to-A). Device D and Device E do not need **neighbor** statements for the client devices (Device B and Device C).



**TIP:** Device D and Device E are considered to be nonclients because they have explicitly configured peer relationships with each other. To make them RRroute reflector clients, remove the **neighbor 192.168.5.5** statement from the configuration on Device D, and remove the **neighbor 192.168.0.1** statement from the configuration on Device E.

---

Figure 50: IBGP Network Using a Route Reflector



### 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 A**

```

set interfaces fe-0/0/0 unit 1 description to-B
set interfaces fe-0/0/0 unit 1 family inet address 10.10.10.1/30
set interfaces fe-0/0/1 unit 3 description to-D
set interfaces fe-0/0/1 unit 3 family inet address 10.10.10.9/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers cluster 192.168.6.5
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.1
set protocols ospf area 0.0.0.0 interface fe-0/0/1.3
set policy-options policy-statement send-ospf term 2 from protocol ospf

```

```
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

**Device B**

```
set interfaces fe-0/0/0 unit 2 description to-A
set interfaces fe-0/0/0 unit 2 family inet address 10.10.10.2/30
set interfaces fe-0/0/1 unit 5 description to-C
set interfaces fe-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.2
set protocols ospf area 0.0.0.0 interface fe-0/0/1.5
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17
```

**Device C**

```
set interfaces fe-0/0/0 unit 6 description to-B
set interfaces fe-0/0/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.6
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

**Device D**

```
set interfaces fe-0/0/0 unit 4 description to-A
set interfaces fe-0/0/0 unit 4 family inet address 10.10.10.10/30
set interfaces fe-0/0/1 unit 7 description to-E
set interfaces fe-0/0/1 unit 7 family inet address 10.10.10.13/30
set interfaces lo0 unit 4 family inet address 192.168.0.1/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.0.1
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.4
set protocols ospf area 0.0.0.0 interface fe-0/0/1.7
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 17
```

**Device E**

```
set interfaces fe-0/0/0 unit 8 description to-D
set interfaces fe-0/0/0 unit 8 family inet address 10.10.10.14/30
```

```

set interfaces lo0 unit 5 family inet address 192.168.5.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.5.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.8
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.5.5
set routing-options autonomous-system 17

```

### Configuring the Route Reflector

**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 IBGP in the network using Juniper Networks Device A as a route reflector:

1. Configure the interfaces.

```

[edit interfaces]
user@A# set fe-0/0/0 unit 1 description to-B
user@A# set fe-0/0/0 unit 1 family inet address 10.10.1/30
user@A# set fe-0/0/1 unit 3 description to-D
user@A# set fe-0/0/1 unit 3 family inet address 10.10.9/30
user@A# set lo0 unit 1 family inet address 192.168.6.5/32

```

2. Configure BGP, including the cluster identifier and neighbor relationships with all IBGP-enabled devices in the autonomous system (AS).

Also apply the policy that redistributes OSPF routes into BGP.

```

[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set local-address 192.168.6.5
user@A# set export send-ospf
user@A# set cluster 192.168.6.5
user@A# set neighbor 192.163.6.4
user@A# set neighbor 192.168.40.4
user@A# set neighbor 192.168.0.1
user@A# set neighbor 192.168.5.5

```

3. Configure static routing or an interior gateway protocol (IGP).

This example uses OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@A# set interface lo0.1 passive
user@A# set interface fe-0/0/0.1
user@A# set interface fe-0/0/1.3

```

4. Configure the policy that redistributes OSPF routes into BGP.

```

[edit policy-options policy-statement send-ospf term 2]
user@A# set from protocol ospf
user@A# set then accept

```

5. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17
```

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

```
user@A# show interfaces
fe-0/0/0 {
  unit 1 {
    description to-B;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
fe-0/0/1 {
  unit 3 {
    description to-D;
    family inet {
      address 10.10.10.9/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}

user@A# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.6.5;
    export send-ospf;
    cluster 192.168.6.5;
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
    neighbor 192.168.0.1;
    neighbor 192.168.5.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-0/0/0.1;
    interface fe-0/0/1.3;
```

```

    }
  }

user@A# show policy-options
policy-statement send-ospf {
  term 2 {
    from protocol ospf;
    then accept;
  }
}

user@A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;

```

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



**NOTE:** Repeat these steps for each nonclient BGP peer within the cluster that you are configuring, if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

### Configuring Client Peers

**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 client peers:

1. Configure the interfaces.
 

```

[edit interfaces]
user@B# set fe-0/0/0 unit 2 description to-A
user@B# set fe-0/0/0 unit 2 family inet address 10.10.10.2/30
user@B# set fe-0/0/1 unit 5 description to-C
user@B# set fe-0/0/1 unit 5 family inet address 10.10.10.5/30
user@B# set lo0 unit 2 family inet address 192.163.6.4/32

```

2. Configure the BGP neighbor relationship with the route reflector.

Also apply the policy that redistributes OSPF routes into BGP.

```

[edit protocols bgp group internal-peers]
user@B# set type internal
user@B# set local-address 192.163.6.4
user@B# set export send-ospf
user@B# set neighbor 192.168.6.5

```

3. Configure OSPF.
 

```

[edit protocols ospf area 0.0.0.0]
user@B# set interface lo0.2 passive
user@B# set interface fe-0/0/0.2
user@B# set interface fe-0/0/1.5

```
4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@B# set from protocol ospf
user@B# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@B# set router-id 192.163.6.4
user@B# set autonomous-system 17
```

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

```
user@B# show interfaces
fe-0/0/0 {
  unit 2 {
    description to-A;
    family inet {
      address 10.10.10.2/30;
    }
  }
}
fe-0/0/1 {
  unit 5 {
    description to-C;
    family inet {
      address 10.10.10.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.163.6.4/32;
    }
  }
}

user@B# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.163.6.4;
    export send-ospf;
    neighbor 192.168.6.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
    interface fe-0/0/0.2;
    interface fe-0/0/1.5;
```

```

    }
  }

user@B# show policy-options
policy-statement send-ospf {
  term 2 {
    from protocol ospf;
    then accept;
  }
}

user@B# show routing-options
router-id 192.163.6.4;
autonomous-system 17;

```

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



**NOTE:** Repeat these steps for each client BGP peer within the cluster that you are configuring if the other client devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

### Configuring Nonclient Peers

**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 nonclient peers:

1. Configure the interfaces.
 

```

[edit interfaces]
user@D# set fe-0/0/0 unit 4 description to-A
user@D# set fe-0/0/0 unit 4 family inet address 10.10.10.10/30
user@D# set fe-0/0/1 unit 7 description to-E
user@D# set fe-0/0/1 unit 7 family inet address 10.10.10.13/30
user@D# set lo0 unit 4 family inet address 192.168.0.1/32
      
```
2. Configure the BGP neighbor relationships with the RRroute reflector and with the other nonclient peers.

Also apply the policy that redistributes OSPF routes into BGP.

```

[edit protocols bgp group internal-peers]
user@D# set type internal
user@D# set local-address 192.168.0.1
user@D# set export send-ospf
user@D# set neighbor 192.168.6.5
user@D# set neighbor 192.168.5.5

```

3. Configure OSPF.
 

```

[edit protocols ospf area 0.0.0.0]
user@D# set interface lo0.4 passive
user@D# set interface fe-0/0/0.4
      
```

```
user@D# set interface fe-0/0/1.7
```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@D# set from protocol ospf
user@D# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@D# set router-id 192.168.0.1
user@D# set autonomous-system 17
```

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

```
user@D# show interfaces
fe-0/0/0 {
  unit 4 {
    description to-A;
    family inet {
      address 10.10.10.10/30;
    }
  }
}
fe-0/0/1 {
  unit 7 {
    description to-E;
    family inet {
      address 10.10.10.13/30;
    }
  }
}
lo0 {
  unit 4 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
```

```
user@D# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.0.1;
    export send-ospf;
    neighbor 192.168.6.5;
    neighbor 192.168.5.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.4 {
```

```

        passive;
    }
    interface fe-0/0/0.4;
    interface fe-0/0/1.7;
}
}

user@D# show policy-options
policy-statement send-ospf {
    term 2 {
        from protocol ospf;
        then accept;
    }
}

user@D# show routing-options
router-id 192.168.0.1;
autonomous-system 17;

```

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



**NOTE:** Repeat these steps for each nonclient BGP peer within the cluster that you are configuring if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

## Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 431](#)
- [Verifying BGP Groups on page 434](#)
- [Verifying BGP Summary Information on page 434](#)
- [Verifying Routing Table Information on page 434](#)

### Verifying BGP Neighbors

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is established for each neighbor address.

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

```

user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+62857 AS 17
  Type: Internal    State: Established (route reflector client)Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4    Local ID: 192.168.6.5    Active Holdtime: 90
  Keepalive Interval: 30    Peer index: 0
  BFD: disabled, down

```

```

NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 17)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        6
  Accepted prefixes:        1
  Suppressed due to damping: 0
  Advertised prefixes:      6
Last traffic (seconds): Received 5    Sent 3    Checked 19
Input messages: Total 2961    Updates 7    Refreshes 0    Octets 56480
Output messages: Total 2945    Updates 6    Refreshes 0    Octets 56235
Output Queue[0]: 0

Peer: 192.168.0.1+179 AS 17    Local: 192.168.6.5+60068 AS 17
Type: Internal    State: Established (route reflector client)Flags: <Sync>
Last State: OpenConfirm    Last Event: RecvKeepAlive
Last Error: None
Export: [ send-ospf ]
Options: <Preference LocalAddress Cluster Refresh>
Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 192.168.0.1    Local ID: 192.168.6.5    Active Holdtime: 90
Keepalive Interval: 30    Peer index: 3
BFD: disabled, down
NLRI for restart configured on peer: inet-unicast
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-unicast
NLRI that restart is negotiated for: inet-unicast
NLRI of received end-of-rib markers: inet-unicast
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 17)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        6
  Accepted prefixes:        1
  Suppressed due to damping: 0
  Advertised prefixes:      6
Last traffic (seconds): Received 18    Sent 20    Checked 12
Input messages: Total 15    Updates 5    Refreshes 0    Octets 447
Output messages: Total 554    Updates 4    Refreshes 0    Octets 32307

```

Output Queue[0]: 0

```

Peer: 192.168.5.5+57458 AS 17 Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established (route reflector client)Flags: <Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.5.5    Local ID: 192.168.6.5    Active Holdtime: 90
  Keepalive Interval: 30    Peer index: 2
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        7
    Accepted prefixes:        7
    Suppressed due to damping: 0
    Advertised prefixes:      6
  Last traffic (seconds): Received 17    Sent 3    Checked 9
  Input messages: Total 2967    Updates 7    Refreshes 0    Octets 56629
  Output messages: Total 2943    Updates 6    Refreshes 0    Octets 56197
  Output Queue[0]: 0

```

```

Peer: 192.168.40.4+53990 AS 17 Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established (route reflector client)Flags: <Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5    Active Holdtime: 90
  Keepalive Interval: 30    Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast

```

```

Peer supports 4 byte AS extension (peer-as 17)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        7
  Accepted prefixes:        7
  Suppressed due to damping: 0
  Advertised prefixes:      6
Last traffic (seconds): Received 5   Sent 23   Checked 52
Input messages: Total 2960   Updates 7   Refreshes 0   Octets 56496
Output messages: Total 2943   Updates 6   Refreshes 0   Octets 56197
Output Queue[0]: 0

```

### Verifying BGP Groups

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From operational mode, enter the **show bgp group** command.

```

user@A> show bgp group
Group Type: Internal   AS: 17                      Local AS: 17
Name: internal-peers  Index: 0                     Flags: <>
Export: [ send-ospf ]
Options: <Cluster>
Holdtime: 0
Total peers: 4          Established: 4
192.163.6.4+179
192.168.40.4+53990
192.168.0.1+179
192.168.5.5+57458
inet.0: 0/26/16/0

Groups: 1  Peers: 4   External: 0   Internal: 4   Down peers: 0   Flaps: 0
Table      Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet.0          26         0         0         0         0         0         0

```

### Verifying BGP Summary Information

**Purpose** Verify that the BGP configuration is correct.

**Action** From operational mode, enter the **show bgp summary** command.

```

user@A> show bgp summary

Groups: 1 Peers: 4 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet.0          26         0         0         0         0         0         0
Peer      AS      InPkt    OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4      17      2981      2965        0        0  22:19:15 0/6/1/0      0/0/0/0
192.168.0.1      17        36        575        0        0   13:43 0/6/1/0      0/0/0/0
192.168.5.5      17      2988      2964        0        0  22:19:10 0/7/7/0      0/0/0/0
192.168.40.4     17      2980      2964        0        0  22:19:14 0/7/7/0      0/0/0/0

```

### Verifying Routing Table Information

**Purpose** Verify that the routing table contains the IBGP routes.

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

```

user@A> show route
inet.0: 12 destinations, 38 routes (12 active, 0 holddown, 10 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      * [Direct/0] 22:22:03
                  > via fe-0/0/0.1
                  [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
                  [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
10.10.10.1/32     * [Local/0] 22:22:03
                  Local via fe-0/0/0.1
10.10.10.4/30     * [OSPF/10] 22:21:13, metric 2
                  > to 10.10.10.2 via fe-0/0/0.1
                  [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
10.10.10.8/30     * [Direct/0] 22:22:03
                  > via fe-0/0/1.3
                  [BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
10.10.10.9/32     * [Local/0] 22:22:03
                  Local via fe-0/0/1.3
10.10.10.12/30    * [OSPF/10] 22:21:08, metric 2
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
192.163.6.4/32    * [OSPF/10] 22:21:13, metric 1
                  > to 10.10.10.2 via fe-0/0/0.1
                  [BGP/170] 22:20:55, MED 1, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
                  [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
192.168.0.1/32    * [OSPF/10] 22:21:08, metric 1
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 22:20:51, MED 1, localpref 100, from 192.168.5.5
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
192.168.5.5/32    * [OSPF/10] 22:21:08, metric 2
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 00:15:24, MED 1, localpref 100, from 192.168.0.1
                  AS path: I
                  > to 10.10.10.10 via fe-0/0/1.3
                  [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                  AS path: I
                  > to 10.10.10.2 via fe-0/0/0.1
192.168.6.5/32    * [Direct/0] 22:22:04

```

```

> via lo0.1
[BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
AS path: I
> to 10.10.10.10 via fe-0/0/1.3
[BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
AS path: I
192.168.40.4/32 > to 10.10.10.2 via fe-0/0/0.1
*[OSPF/10] 22:21:13, metric 2
> to 10.10.10.2 via fe-0/0/0.1
[BGP/170] 22:20:55, MED 1, localpref 100, from 192.163.6.4
AS path: I
> to 10.10.10.2 via fe-0/0/0.1
[BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
AS path: I
224.0.0.5/32 > to 10.10.10.10 via fe-0/0/1.3
*[OSPF/10] 22:22:07, metric 1
MultiRecv

```

- Related Documentation**
- [Understanding External BGP Peering Sessions on page 33](#)
  - [BGP Configuration Overview on page 12](#)

## Example: Configuring a Route Reflector that Belongs to Two Different Clusters

- [Understanding a Route Reflector that Belongs to Two Different Clusters on page 436](#)
- [Example: Configuring a Route Reflector that Belongs to Two Different Clusters on page 437](#)

### Understanding a Route Reflector that Belongs to Two Different Clusters

The purpose of route reflection is loop prevention when the internal BGP (IBGP) routing devices are not fully meshed. To accomplish this, RRs break one of the rules of normal BGP operation: They readvertise routes learned from an internal BGP peer to other internal BGP peers.

Normally, a single RR is a member of only one cluster. Consider, for example, that in a hierarchical RR design, a tier-two RR can be the client of a tier-1 RR, but they can not be clients of each other.

However, when two RRs are clients of each other and the routes are being reflected from one cluster to another, only one of the cluster IDs is included in the cluster list. This is because having one cluster ID in the cluster list is adequate for loop prevention in this case.

[Table 5 on page 437](#) summarizes the rules that route reflectors use when filling in a reflected route's cluster list with cluster IDs.

Table 5: Rules for Route Reflectors

Route Reflection Scenario	Configuration
When reflecting a route from one of the clients to a non-client router  client -> RR -> non-client	The RR fills the cluster ID associated with that client in the cluster list of the reflected route.
When reflecting a route from a non-client router to a client router  non-client -> RR -> client	The RR fills the cluster ID associated with that client in the cluster list of the reflected route.
When reflecting a route from a client router to another client router that is in a different cluster  client1 -> RR -> client2 (different cluster)	The RR fills the cluster ID associated with client1 in the cluster list before reflecting the cluster ID to client2. The cluster ID associated with client 2 is not added.

### Example: Configuring a Route Reflector that Belongs to Two Different Clusters

This example shows how to configure a route reflector (RRs) that belongs to two different clusters. This is not a common scenario, but it might be useful in some situations.

- [Requirements on page 437](#)
- [Overview on page 437](#)
- [Configuration on page 438](#)
- [Verification on page 440](#)

#### Requirements

Configure the device interfaces and an internal gateway protocol (IGP). For an example of an RR setup that includes the interface and IGP configuration, see [“Example: Configuring a Route Reflector” on page 421](#).

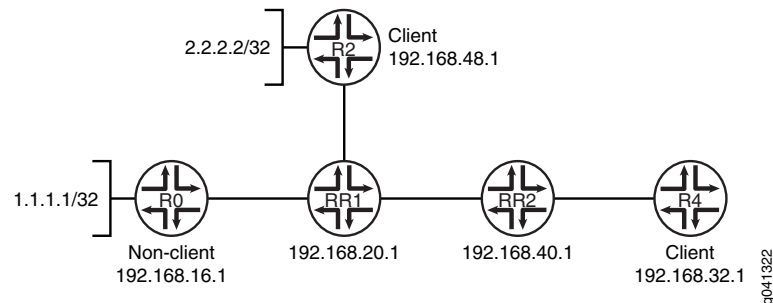
#### Overview

In this example, Device RR1 is a route reflector for both Device R2 and Device RR2.

Device RR2 is a route reflector for Device R4.

Consider figure [Figure 51 on page 438](#).

Figure 51: Route Reflector in Two Different Clusters



This example shows the BGP configuration on Device RR1 and Device RR2.

### 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 RR1
set protocols bgp group RR1_client type internal
set protocols bgp group RR1_client local-address 192.168.20.1
set protocols bgp group RR1_client cluster 5.5.5.5
set protocols bgp group RR1_client neighbor 192.168.48.1
set protocols bgp group Non_client type internal
set protocols bgp group Non_client local-address 192.168.20.1
set protocols bgp group Non_client neighbor 192.168.16.1
set protocols bgp group RR1_to_RR2 type internal
set protocols bgp group RR1_to_RR2 local-address 192.168.20.1
set protocols bgp group RR1_to_RR2 cluster 6.6.6.6
set protocols bgp group RR1_to_RR2 neighbor 192.168.40.1

Device RR2
set protocols bgp group RR2_client type internal
set protocols bgp group RR2_client local-address 192.168.40.1
set protocols bgp group RR2_client cluster 7.7.7.7
set protocols bgp group RR2_client neighbor 192.168.32.1
set protocols bgp group RR2_to_RR1 type internal
set protocols bgp group RR2_to_RR1 local-address 192.168.40.1
set protocols bgp group RR2_to_RR1 neighbor 192.168.20.1

```

### Configuring Device RR1

#### 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 Device RR1:

1. Configure the peering relationship with Device R2.

```

[edit protocols bgp group RR1_client]
user@RR1# set type internal
user@RR1# set local-address 192.168.20.1
user@RR1# set cluster 5.5.5.5

```

```
user@RR1# set neighbor 192.168.48.1
```

2. Configure the peering relationship with Device R0.

```
[edit protocols bgp group Non_client]
user@RR1# set type internal
user@RR1# set local-address 192.168.20.1
user@RR1# set neighbor 192.168.16.1
```

3. Configure the peering relationship with Device RR2.

```
[edit protocols bgp group RR1_to_RR2]
user@RR1# set type internal
user@RR1# set local-address 192.168.20.1
user@RR1# set cluster 6.6.6.6
user@RR1# set neighbor 192.168.40.1
```

**Results** From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@RR1# show protocols
bgp {
  group RR1_client {
    type internal;
    local-address 192.168.20.1;
    cluster 5.5.5.5;
    neighbor 192.168.48.1;
  }
  group Non_client {
    type internal;
    local-address 192.168.20.1;
    neighbor 192.168.16.1;
  }
  group RR1_to_RR2 {
    type internal;
    local-address 192.168.20.1;
    cluster 6.6.6.6;
    neighbor 192.168.40.1;
  }
}
```

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

### **Configuring Device RR2**

**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 Device RR2:

1. Configure the peering relationship with Device R2.

```
[edit protocols bgp group RR2_client]
user@RR2# set type internal
user@RR2# set local-address 192.168.40.1
```

```
user@RR2# set cluster 7.7.7.7
user@RR2# set neighbor 192.168.32.1
```

2. Configure the peering relationship with Device R0.

```
[edit protocols bgp group RR2_to_RR1]
user@RR2# set type internal
user@RR2# set local-address 192.168.40.1
user@RR2# set neighbor 192.168.20.1
```

**Results** From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@RR2# show protocols
bgp {
  group RR2_client {
    type internal;
    local-address 192.168.40.1;
    cluster 7.7.7.7;
    neighbor 192.168.32.1;
  }
  group RR2_to_RR1 {
    type internal;
    local-address 192.168.40.1;
    neighbor 192.168.20.1;
  }
}
```

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

---

### Verification

Confirm that the configuration is working properly.

- [Checking the Cluster ID Advertised for Route 2.2.2.2 on page 440](#)
- [Checking the Cluster ID Advertised for Route 1.1.1.1 on page 441](#)

#### *Checking the Cluster ID Advertised for Route 2.2.2.2*

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is established for each neighbor address.

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

```
user@RR1> show route advertising-protocol bgp 192.168.40.1 active-path 2.2.2.2 extensive
```

```
inet.0: 61 destinations, 61 routes (60 active, 0 holddown, 1 hidden)
* 2.2.2.2/32 (1 entry, 1 announced)
  BGP group RR1_to_RR2 type Internal
    Nexthop: 192.168.48.1
    Localpref: 100
    AS path: [100] I
    Cluster ID: 5.5.5.5
    Originator ID: 192.168.48.1
```

**Meaning** The 2.2.2.2/32 route originates from the Device RR1's client peer, Device R2. When this route is sent to RR1's client, Device RR2, the route has the 5.5.5.5 cluster ID attached, which is the cluster ID for RR1-RR2.

#### *Checking the Cluster ID Advertised for Route 1.1.1.1*

**Purpose** Check the route advertisement from Device RR1 to Device RR2.

**Action** From operational mode, enter the **show bgp group** command.

```
user@RR1> show route advertising-protocol bgp 192.168.40.1 active-path 1.1.1.1/32 extensive
inet.0: 61 destinations, 61 routes (60 active, 0 holddown, 1 hidden)
* 1.1.1.1/32 (1 entry, 1 announced)
  BGP group RR1_to_RR2 type Internal
    Nexthop: 192.168.16.1
    Localpref: 100
    AS path: [100] I
    Cluster ID: 6.6.6.6
    Originator ID: 192.168.16.1
```

**Meaning** The 1.1.1.1/32 route originates from the Device RR1's non-client peer, Device R0. When this route is sent to RR1's client, Device RR2, the route has the 6.6.6.6 cluster ID attached, which is the cluster ID for RR1-RR2.

Device RR1 preserves the inbound cluster ID from Device R2 when advertising to another client in a different cluster (Device R4).

**Related Documentation**

- [Example: Configuring BGP Route Reflectors on page 419](#)

## Example: Configuring BGP Confederations

- [Understanding BGP Confederations on page 441](#)
- [Example: Configuring BGP Confederations on page 442](#)

### Understanding BGP Confederations

BGP confederations are another way to solve the scaling problems created by the BGP full mesh requirement. BGP confederations effectively break up a large autonomous system (AS) into subautonomous systems (sub-ASs). Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64,512 and 65,535.

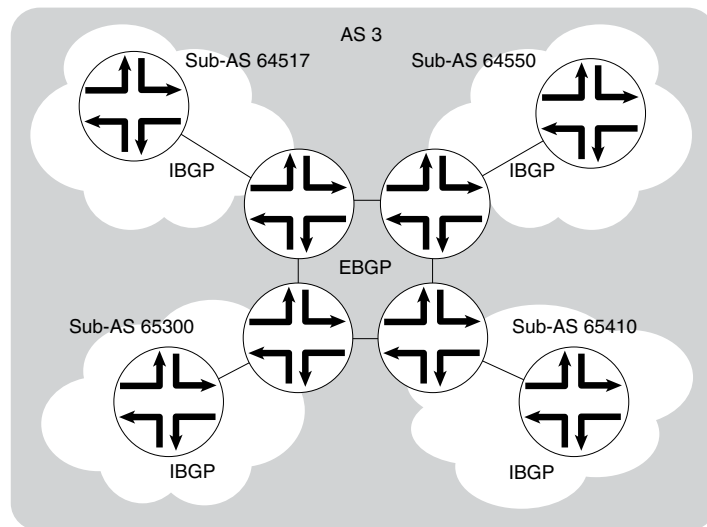
Within a sub-AS, the same internal BGP (IBGP) full mesh requirement exists. Connections to other confederations are made with standard external BGP (EBGP), and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

The confederation AS appears whole to other confederation ASs. The AS path received by other ASs shows only the globally assigned AS number. It does not include the confederation sequence or the privately assigned sub-AS numbers. The sub-AS numbers

are removed when the route is advertised out of the confederation AS.

[Figure 52 on page 442](#) shows an AS divided into four confederations.

**Figure 52: BGP Confederations**



[Figure 52 on page 442](#) shows AS 3 divided into four sub-ASs, 64517, 64550, 65300, and 65410, which are linked through EBGP sessions. Because the confederations are connected by EBGP, they do not need to be fully meshed. EBGP routes are readvertised to other sub-ASs.

### Example: Configuring BGP Confederations

This example shows how to configure BGP confederations.

- [Requirements on page 442](#)
- [Overview on page 442](#)
- [Configuration on page 443](#)
- [Verification on page 445](#)

#### Requirements

- Configure network interfaces.
- Configure external peer sessions. See [“Example: Configuring External BGP Point-to-Point Peer Sessions” on page 34](#).
- Configure interior gateway protocol (IGP) sessions between peers.
- Configure a routing policy to advertise the BGP routes.

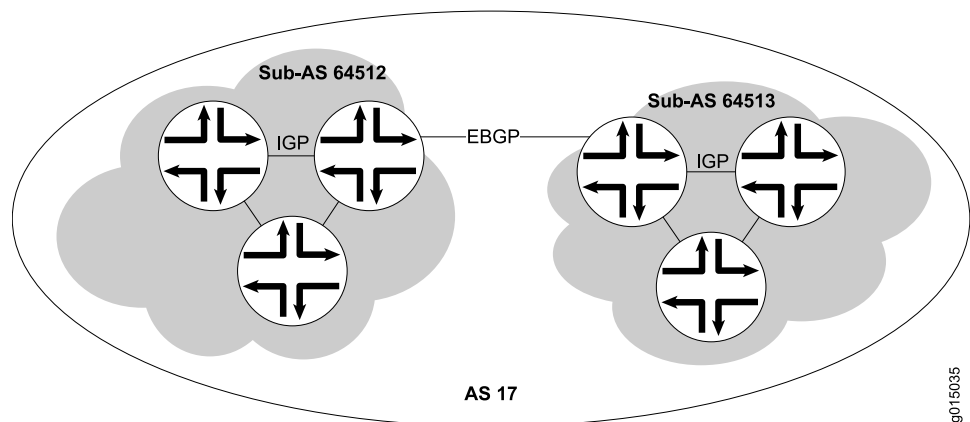
#### Overview

Within a BGP confederation, the links between the confederation member autonomous systems (ASs) must be external BGP (EBGP) links, not internal BGP (IBGP) links.

Similar to route reflectors, BGP confederations reduce the number of peer sessions and TCP sessions to maintain connections between IBGP routing devices. BGP confederation is one method used to solve the scaling problems created by the IBGP full mesh requirement. BGP confederations effectively break up a large AS into subautonomous systems. Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64512 and 65535. Within a sub-AS, the same IBGP full mesh requirement exists. Connections to other confederations are made with standard EBGP, and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

Figure 53 on page 443 shows a sample network in which AS 17 has two separate confederations: sub-AS 64512 and sub-AS 64513, each of which has multiple routers. Within a sub-AS, an IGP is used to establish network connectivity with internal peers. Between sub-ASs, an EBGP peer session is established.

**Figure 53: Typical Network Using BGP Confederations**



### 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.

#### All Devices in Sub-AS 64512

```
set routing-options autonomous-system 64512
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64512 type internal
set protocols bgp group sub-AS-64512 local-address 192.168.5.1
set protocols bgp group sub-AS-64512 neighbor 192.168.8.1
set protocols bgp group sub-AS-64512 neighbor 192.168.15.1
```

#### Border Device in Sub-AS 64512

```
set protocols bgp group to-sub-AS-64513 type external
set protocols bgp group to-sub-AS-64513 peer-as 64513
set protocols bgp group to-sub-AS-64513 neighbor 192.168.5.2
```

**All Devices in Sub-AS 64513**

```
set routing-options autonomous-system 64513
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64513 type internal
set protocols bgp group sub-AS-64513 local-address 192.168.5.2
set protocols bgp group sub-AS-64513 neighbor 192.168.9.1
set protocols bgp group sub-AS-64513 neighbor 192.168.16.1
```

**Border Device in Sub-AS 64513**

```
set protocols bgp group to-sub-AS-64512 type external
set protocols bgp group to-sub-AS-64512 peer-as 64512
set protocols bgp group to-sub-AS-64512 neighbor 192.168.5.1
```

**Step-by-Step Procedure**

This procedure shows the steps for the devices that are in sub-AS 64512.

The **autonomous-system** statement sets the sub-AS number of the device.

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 BGP confederations:

1. Set the sub-AS number for the device.

```
[edit routing-options]
user@host# set autonomous-system 64512
```

2. In the confederation, include all sub-ASs in the main AS.

The number 17 represents the main AS. The **members** statement lists all the sub-ASs in the main AS.

```
[edit routing-options confederation]
user@host# set 17 members 64512
user@host# set 17 members 64513
```

3. On the border device in sub-AS 64512, configure an EBGP connection to the border device in AS 64513.

```
[edit protocols bgp group to-sub-AS-64513]
user@host# set type external
user@host# set neighbor 192.168.5.2
user@host# set peer-as 64513
```

4. Configure an IBGP group for peering with the devices within sub-AS 64512.

```
[edit protocols bgp group sub-AS-64512]
user@host# set type internal
user@host# set local-address 192.168.5.1
user@host# neighbor 192.168.8.1
user@host# neighbor 192.168.15.1
```

**Results** From configuration mode, confirm your configuration by entering the **show routing-options** 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 routing-options
```

```

autonomous-system 64512;
confederation 17 members [ 64512 64513 ];

user@host# show protocols
bgp {
  group to-sub-AS-64513 { # On the border devices only
    type external;
    peer-as 64513;
    neighbor 192.168.5.2;
  }
  group sub-AS-64512 {
    type internal;
    local-address 192.168.5.1;
    neighbor 192.168.8.1;
    neighbor 192.168.15.1;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for sSub-AS 64513.

### Verification

Confirm that the configuration is working properly.

- [Verifying BGP Neighbors on page 445](#)
- [Verifying BGP Groups on page 446](#)
- [Verifying BGP Summary Information on page 447](#)

#### *Verifying BGP Neighbors*

**Purpose** Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

**Action** From the CLI, enter the **show bgp neighbor** command.

### Sample Output

```

user@host> show bgp neighbor
Peer: 10.255.245.12+179 AS 35 Local: 10.255.245.13+2884 AS 35
  Type: Internal State: Established (route reflector client)Flags: Sync
  Last State: OpenConfirm Last Event: RecvKeepAlive
  Last Error: None
  Options: Preference LocalAddress HoldTime Cluster AddressFamily Rib-group Refresh

  Address families configured: inet-vpn-unicast inet-labeled-unicast
  Local Address: 10.255.245.13 Holdtime: 90 Preference: 170
  Flags for NLRI inet-vpn-unicast: AggregateLabel
  Flags for NLRI inet-labeled-unicast: AggregateLabel
  Number of flaps: 0
  Peer ID: 10.255.245.12 Local ID: 10.255.245.13 Active Holdtime: 90
  Keepalive Interval: 30
  NLRI advertised by peer: inet-vpn-unicast inet-labeled-unicast
  NLRI for this session: inet-vpn-unicast inet-labeled-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 300
  Stale routes from peer are kept for: 60

```

```

Restart time requested by this peer: 300
NLRI that peer supports restart for: inet-unicast inet6-unicast
NLRI that restart is negotiated for: inet-unicast inet6-unicast
NLRI of received end-of-rib markers: inet-unicast inet6-unicast
NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
Table inet.0 Bit: 10000
  RIB State: restart is complete
  Send state: in sync
  Active prefixes: 4
  Received prefixes: 6
  Suppressed due to damping: 0
Table inet6.0 Bit: 20000
  RIB State: restart is complete
  Send state: in sync
  Active prefixes: 0
  Received prefixes: 2
  Suppressed due to damping: 0
Last traffic (seconds): Received 3    Sent 3    Checked 3
Input messages: Total 9    Updates 6    Refreshes 0    Octets 403
Output messages: Total 7    Updates 3    Refreshes 0    Octets 365
Output Queue[0]: 0
Output Queue[1]: 0
Trace options: detail packets
Trace file: /var/log/bgpr size 131072 files 10

```

**Meaning** The output shows a list of the BGP neighbors with detailed session information. Verify the following information:

- Each configured peering neighbor is listed.
- For **State**, each BGP session is **Established**.
- For **Type**, each peer is configured as the correct type (either internal or external).
- For **AS**, the AS number of the BGP neighbor is correct.

#### Verifying BGP Groups

**Purpose** Verify that the BGP groups are configured correctly.

**Action** From the CLI, enter the **show bgp group** command.

### Sample Output

```

user@host> show bgp group
Group Type: Internal  AS: 10045      Local AS: 10045
Name: pe-to-asbr2                      Flags: Export Eval
Export: [ match-all ]
Total peers: 1      Established: 1
10.0.0.4+179
bgp.l3vpn.0: 1/1/0
vpn-green.inet.0: 1/1/0

Groups: 1  Peers: 1  External: 0  Internal: 1  Down peers: 0  Flaps: 0
Table      Tot Paths  Act Paths  Suppressed  History Damp State  Pending
bgp.l3vpn.0      1          1          0          0          0          0

```

**Meaning** The output shows a list of the BGP groups with detailed group information. Verify the following information:

- Each configured group is listed.
- For **AS**, each group's remote AS is configured correctly.
- For **Local AS**, each group's local AS is configured correctly.
- For **Group Type**, each group has the correct type (either internal or external).
- For **Total peers**, the expected number of peers within the group is shown.
- For **Established**, the expected number of peers within the group have BGP sessions in the **Established** state.
- The IP addresses of all the peers within the group are present.

### Verifying BGP Summary Information

**Purpose** Verify that the BGP configuration is correct.

**Action** From the CLI, enter the **show bgp summary** command.

### Sample Output

```
user@host> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History  Damp State   Pending
inet.0           6        4          0          0      0      0
Peer      AS      InPkt    OutPkt    OutQ    Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2    65002    88675    88652      0        2      42:38 2/4/0
           0/0/0
10.0.0.3    65002    54528    54532      0        1    2w4d22h 0/0/0
           0/0/0
10.0.0.4    65002    51597    51584      0        0    2w3d22h 2/2/0
           0/0/0
```

**Meaning** The output shows a summary of BGP session information. Verify the following information:

- For **Groups**, the total number of configured groups is shown.
- For **Peers**, the total number of BGP peers is shown.
- For **Down Peers**, the total number of unestablished peers is 0. If this value is not zero, one or more peering sessions are not yet established.
- Under **Peer**, the IP address for each configured peer is shown.
- Under **AS**, the peer AS for each configured peer is correct.
- Under **Up/Dwn State**, the BGP state reflects the number of paths received from the neighbor, the number of these paths that have been accepted, and the number of routes being damped (such as 0/0/0). If the field is **Active**, it indicates a problem in the establishment of the BGP session.

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)



## CHAPTER 10

# BGP Security Configuration

- [Example: Configuring BGP Route Authentication on page 449](#)
- [Example: Configuring IPsec Protection for BGP on page 456](#)
- [Examples: Configuring TCP and BGP Security on page 459](#)
- [Example: Configuring Origin Validation for BGP on page 473](#)

### Example: Configuring BGP Route Authentication

---

- [Understanding Route Authentication on page 449](#)
- [Example: Configuring Route Authentication for BGP on page 450](#)

### Understanding Route Authentication

The use of router and route authentication and route integrity greatly mitigates the risk of being attacked by a machine or router that has been configured to share incorrect routing information with another router. In this kind of attack, the attacked router can be tricked into creating a routing loop, or the attacked router's routing table can be greatly increased thus impacting performance, or routing information can be redirected to a place in the network for the attacker to analyze it. Bogus route advertisements can be sent out on a segment. These updates can be accepted into the routing tables of neighbor routers unless an authentication mechanism is in place to verify the source of the routes.

Router and route authentication enables routers to share information only if they can verify that they are talking to a trusted source, based on a password (key). In this method, a hashed key is sent along with the route being sent to another router. The receiving router compares the sent key to its own configured key. If they are the same, it accepts the route. By using a hashing algorithm, the key is not sent over the wire in plain text. Instead, a hash is calculated using the configured key. The routing update is used as the input text, along with the key, into the hashing function. This hash is sent along with the route update to the receiving router. The receiving router compares the received hash with a hash it generates on the route update using the preshared key configured on it. If the two hashes are the same, the route is assumed to be from a trusted source. The key is known only to the sending and receiving routers.

To further strengthen security, you can configure a series of authentication keys (a *keychain*). Each key has a unique start time within the keychain. Keychain authentication allows you to change the password information periodically without bringing down peering sessions. This keychain authentication method is referred to as *hitless* because

the keys roll over from one to the next without resetting any peering sessions or interrupting the routing protocol.

The sending peer uses the following rules to identify the active authentication key:

- The start time is less than or equal to the current time (in other words, not in the future).
- The start time is greater than that of all other keys in the chain whose start time is less than the current time (in other words, closest to the current time).

The receiving peer determines the key with which it authenticates based on the incoming key identifier.

The sending peer identifies the current authentication key based on a configured start time and then generates a hash value using the current key. The sending peer then inserts a TCP-enhanced authentication option object into the BGP update message. The object contains an object ID (assigned by IANA), the object length, the current key, and a hash value.

The receiving peer examines the incoming TCP-enhanced authentication option, looks up the received authentication key, and determines whether the key is acceptable based on the start time, the system time, and the tolerance parameter. If the key is accepted, the receiving peer calculates a hash and authenticates the update message.

Initial application of a keychain to a TCP session causes the session to reset. However, once the keychain is applied, the addition or removal of a password from the keychain does not cause the TCP session to reset. Also, the TCP session does not reset when the keychain changes from one authentication algorithm to another.

## Example: Configuring Route Authentication for BGP

All BGP protocol exchanges can be authenticated to guarantee that only trusted routing devices participate in autonomous system (AS) routing updates. By default, authentication is disabled.

- [Requirements on page 450](#)
- [Overview on page 450](#)
- [Configuration on page 452](#)
- [Verification on page 454](#)

---

### Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol (IGP).

---

### Overview

When you configure authentication, the algorithm creates an encoded checksum that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet's checksum.

This example includes the following statements for configuring and applying the keychain:

- **key**—A keychain can have multiple keys. Each key within a keychain must be identified by a unique integer value. The range of valid identifier values is from 0 through 63.  
The key can be up to 126 characters long. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (" ").
- **tolerance**—(Optional) For each keychain, you can configure a clock-skew tolerance value in seconds. The clock-skew tolerance is applicable to the receiver accepting keys for BGP updates. The configurable range is 0 through 999,999,999 seconds. During the tolerance period, either the current or previous password is acceptable.
- **key-chain**—For each keychain, you must specify a name. This example defines one keychain: **bgp-auth**. You can have multiple keychains on a routing device. For example, you can have a keychain for BGP, a keychain for OSPF, and a keychain for LDP.
- **secret**—For each key in the keychain, you must set a secret password. This password can be entered in either encrypted or plain text format in the **secret** statement. It is always displayed in encrypted format.
- **start-time**—Each key must specify a start time in UTC format. Control gets passed from one key to the next. When a configured start time arrives (based on the routing device's clock), the key with that start time becomes active. Start times are specified in the local time zone for a routing device and must be unique within the keychain.
- **authentication-key-chain**—Enables you to apply a keychain at the global BGP level for all peers, for a group, or for a neighbor. This example applies the keychain to the peers defined in the external BGP (EBGP) group called **ext**.
- **authentication-algorithm**—For each keychain, you can specify a hashing algorithm. The algorithm can be AES-128, MD5, or SHA-1.

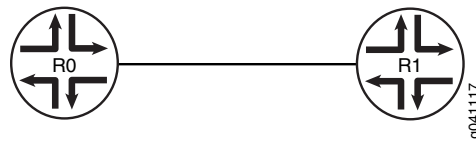
You associate a keychain and an authentication algorithm with a BGP neighboring session.

This example configures a keychain named **bgp-auth**. Key 0 will be sent and accepted starting at 2011-6-23.20:19:33 -0700, and will stop being sent and accepted when the next key in the keychain (key 1) becomes active. Key 1 becomes active one year later at 2012-6-23.20:19:33 -0700, and will not stop being sent and accepted unless another key is configured with a start time that is later than the start time of key 1. A clock-skew tolerance of 30 seconds applies to the receiver accepting the keys. During the tolerance period, either the current or previous key is acceptable. The keys are shared-secret passwords. This means that the neighbors receiving the authenticated routing updates must have the same authentication keychain configuration, including the same keys (passwords). So Router R0 and Router R1 must have the same authentication-key-chain configuration if they are configured as peers. This example shows the configuration on only one of the routing devices.

### **Topology Diagram**

Figure 54 on page 452 shows the topology used in this example.

Figure 54: Authentication for BGP



### 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.

```
set protocols bgp group ext type external
set protocols bgp group ext peer-as 65530
set protocols bgp group ext neighbor 172.16.2.1
set routing-options autonomous-system 65533
set protocols bgp group ext authentication-key-chain bgp-auth
set protocols bgp group ext authentication-algorithm md5
set security authentication-key-chains key-chain bgp-auth tolerance 30
set security authentication-key-chains key-chain bgp-auth key 0 secret
  this-is-the-secret-password
set security authentication-key-chains key-chain bgp-auth key 0 start-time
  2011-6-23.20:19:33-0700
set security authentication-key-chains key-chain bgp-auth key 1 secret
  this-is-another-secret-password
set security authentication-key-chains key-chain bgp-auth key 1 start-time
  2012-6-23.20:19:33-0700
```

#### 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 Router R1 to accept route filters from Device CE1 and perform outbound route filtering using the received filters:

1. Configure the local autonomous system.
 

```
[edit routing-options]
user@R1# set autonomous-system 65533
```
2. Configure one or more BGP groups.
 

```
[edit protocols bgp group ext]
user@R1# set type external
user@R1# set peer-as 65530
user@R1# set neighbor 172.16.2.1
```
3. Configure authentication with multiple keys.
 

```
[edit security authentication-key-chains key-chain bgp-auth]
user@R1# set key 0 secret this-is-the-secret-password
user@R1# set key 0 start-time 2011-6-23.20:19:33-0700
user@R1# set key 1 secret this-is-another-secret-password
user@R1# set key 1 start-time 2012-6-23.20:19:33-0700
```

The start time of each key must be unique within the keychain.

4. Apply the authentication keychain to BGP, and set the hashing algorithm.

```
[edit protocols bgp group ext]
user@R1# set authentication-key-chain bgp-auth
user@R1# set authentication-algorithm md5
```

5. (Optional) Apply a clock-skew tolerance value in seconds.

```
[edit security authentication-key-chains key-chain bgp-auth]
user@R1# set tolerance 30
```

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

```
user@R1# show protocols
bgp {
  group ext {
    type external;
    peer-as 65530;
    neighbor 172.16.2.1;
    authentication-key-chain bgp-auth;
    authentication-algorithm md5;
  }
}

user@R1# show routing-options
autonomous-system 65533;

user@R1# show security
authentication-key-chains {
  key-chain bgp-auth {
    tolerance 30;
    key 0 {
      secret
        "$9$5T6AREYk8RhXNdwaJn/CtO1cykWx9AyIMWdVgoJDjqP5FCA0z3IEhcMWLxNbgJDiF6A";
      ## SECRET-DATA
      start-time "2011-6-23.20:19:33 -0700";
    }
    key 1 {
      secret "$9$UyD.59CuO1h9AylKW-dqmfT369CuRhSP5hrvMN-JGDiqfu0lleWpuh.";
      ## SECRET-DATA
      start-time "2012-6-23.20:19:33 -0700";
    }
  }
}
```

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

Repeat the procedure for every BGP-enabled device in the network, using the appropriate interface names and addresses for each BGP-enabled device.

## Verification

Confirm that the configuration is working properly.

- [Verifying Authentication for the Neighbor on page 454](#)
- [Verifying That Authorization Messages Are Sent on page 454](#)
- [Checking Authentication Errors on page 455](#)
- [Verifying the Operation of the Keychain on page 455](#)

### *Verifying Authentication for the Neighbor*

**Purpose** Make sure that the **AuthKeyChain** option appears in the output of the **show bgp neighbor** command.

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

```
user@R1> show bgp neighbor
Peer: 172.16.2.1+179 AS 65530 Local: 172.16.2.2+1222 AS 65533
  Type: External State: Established Flags: <Sync>
  Last State: OpenConfirm Last Event: RecvKeepAlive
  Last Error: None
  Export: [ direct-lo0 ]
  Options: <Preference PeerAS Refresh>
  Options: <AuthKeyChain>
  Authentication key is configured
  Authentication key chain: jni
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 172.16.2.1 Local ID: 10.255.124.35 Active Holdtime: 90
  Keepalive Interval: 30 Peer index: 0
  Local Interface: fe-0/0/1.0
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes: 2
    Received prefixes: 2
    Suppressed due to damping: 0
    Advertised prefixes: 1
  Last traffic (seconds): Received 2 Sent 2 Checked 2
  Input messages: Total 21 Updates 2 Refreshes 0 Octets 477
  Output messages: Total 22 Updates 1 Refreshes 0 Octets 471
  Output Queue[0]: 0
```

### *Verifying That Authorization Messages Are Sent*

**Purpose** Confirm that BGP has the enhanced authorization option.

**Action** From operational mode, enter the **monitor traffic interface fe-0/0/1** command.

```
user@R1> monitor traffic interface fe-0/0/1
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Listening on fe-0/0/1, capture size 96 bytes
```

```
13:08:00.618402 In arp who-has 172.16.2.66 tell 172.16.2.69
```

```

13:08:02.408249 Out IP 172.16.2.2.1122 > 172.16.2.1.646: P
1889289217:1889289235(18) ack 2215740969 win 58486 <nop,nop,timestamp 167557
1465469,nop,Enhanced Auth keyid 0 diglen 12 digest: fe3366001f45767165f17037>:
13:08:02.418396 In IP 172.16.2.1.646 > 172.16.2.2.1122: P 1:19(18) ack 18 win
57100 <nop,nop,timestamp 1466460 167557,nop,Enhanced Auth keyid 0 diglen 12
digest: a18c31eda1b14b2900921675>:
13:08:02.518146 Out IP 172.16.2.2.1122 > 172.16.2.1.646: . ack 19 win 58468
<nop,nop,timestamp 167568 1466460,nop,Enhanced Auth keyid 0 diglen 12 digest:
c3b6422eb6bd3fd9cf79742b>
13:08:28.199557 Out IP 172.16.2.2.nerv > 172.16.2.1.bgp: P
286842489:286842508(19) ack 931203976 win 57200 <nop,Enhanced Auth keyid 0
diglen 12 digest: fc0e42900a73736bcc07c1a4>: BGP, length: 19
13:08:28.209661 In IP 172.16.2.1.bgp > 172.16.2.2.nerv: P 1:20(19) ack 19 win
56835 <nop,Enhanced Auth keyid 0 diglen 12 digest: 0fc8578c489fabce63aeb2c3>:
BGP, length: 19
13:08:28.309525 Out IP 172.16.2.2.nerv > 172.16.2.1.bgp: . ack 20 win 57181
<nop,Enhanced Auth keyid 0 diglen 12 digest: ef03f282fb2ece0039491df8>
13:08:32.439708 Out IP 172.16.2.2.1122 > 172.16.2.1.646: P 54:72(18) ack 55 win
58432 <nop,nop,timestamp 170560 1468472,nop,Enhanced Auth keyid 0 diglen 12
digest: 76e0cf926f348b726c631944>:
13:08:32.449795 In IP 172.16.2.1.646 > 172.16.2.2.1122: P 55:73(18) ack 72 win
57046 <nop,nop,timestamp 1469463 170560,nop,Enhanced Auth keyid 0 diglen 12
digest: dae3eec390d18a114431f4d8>:
13:08:32.549726 Out IP 172.16.2.2.1122 > 172.16.2.1.646: . ack 73 win 58414
<nop,nop,timestamp 170571 1469463,nop,Enhanced Auth keyid 0 diglen 12 digest:
851df771aee2ea7a43a0c46c>
13:08:33.719880 In arp who-has 172.16.2.66 tell 172.16.2.69
^C
35 packets received by filter
0 packets dropped by kernel

```

### Checking Authentication Errors

**Purpose** Check the number of packets dropped by TCP because of authentication errors.

**Action** From operational mode, enter the **show system statistics tcp | match auth** command.

```

user@R1> show system statistics tcp | match auth
      0 send packets dropped by TCP due to auth errors
      58 rcv packets dropped by TCP due to auth errors

```

### Verifying the Operation of the Keychain

**Purpose** Check the number of packets dropped by TCP because of authentication errors.

**Action** From operational mode, enter the **show security keychain detail** command.

```

user@R1> show security keychain detail
keychain              Active-ID      Next-ID      Transition  Tolerance
                     Send Receive  Send Receive
bgp-auth              3      3      1      1      1d 23:58      30
  Id 3, Algorithm hmac-md5, State send-receive, Option basic
  Start-time Wed Aug 11 16:28:00 2010, Mode send-receive
  Id 1, Algorithm hmac-md5, State inactive, Option basic
  Start-time Fri Aug 20 11:30:57 2010, Mode send-receive

```

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Configuring IPsec Protection for BGP

---

- [Understanding IPsec for BGP on page 456](#)
- [Example: Using IPsec to Protect BGP Traffic on page 456](#)

### Understanding IPsec for BGP

You can apply the IP security (IPsec) to BGP traffic. IPsec is a protocol suite used for protecting IP traffic at the packet level. IPsec is based on security associations (SAs). An SA is a simplex connection that provides security services to the packets carried by the SA. After configuring the SA, you can apply it to BGP peers.

The Junos OS implementation of IPsec supports two types of security: host to host and gateway to gateway. Host-to-host security protects BGP sessions with other routers. An SA to be used with BGP must be configured manually and use transport mode. Static values must be configured on both ends of the security association. To apply host protection, you configure manual SAs in transport mode and then reference the SA by name in the BGP configuration to protect a session with a given peer.

Manual SAs require no negotiation between the peers. All values, including the keys, are static and specified in the configuration. Manual SAs statically define the security parameter index values, algorithms, and keys to be used and require matching configurations on both end points of the tunnel (on both peers). As a result, each peer must have the same configured options for communication to take place.

In transport mode, IPsec headers are inserted after the original IP header and before the transport header.

The security parameter index is an arbitrary value used in combination with a destination address and a security protocol to uniquely identify the SA.

### Example: Using IPsec to Protect BGP Traffic

IPsec is a suite of protocols used to provide secure network connections at the IP layer. It is used to provide data source authentication, data integrity, confidentiality and packet replay protection. This example shows how to configure IPsec functionality to protect Routing Engine-to-Routing Engine BGP sessions. Junos OS supports IPsec Authentication Header (AH) and Encapsulating Security Payload (ESP) in transport and tunnel mode, as well as a utility for creating policies and manually configuring keys.

- [Requirements on page 456](#)
- [Overview on page 457](#)
- [Configuration on page 457](#)
- [Verification on page 459](#)

#### Requirements

---

Before you begin:

- Configure the router interfaces.

- Configure an interior gateway protocol (IGP).
- Configure BGP.

No specific PIC hardware is required to configure this feature.

### Overview

The SA is configured at the `[edit security ipsec security-association name]` hierarchy level with the `mode` statement set to transport. In transport mode, Junos OS does not support authentication header (AH) or encapsulating security payload (ESP) header bundles. Junos OS supports only the BGP protocol in transport mode.

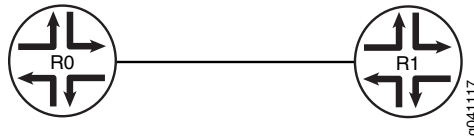
This example specifies bidirectional IPsec to decrypt and authenticate the incoming and outgoing traffic using the same algorithm, keys, and SPI in both directions, unlike inbound and outbound SAs that use different attributes in both directions.

A more specific SA overrides a more general SA. For example, if a specific SA is applied to a specific peer, that SA overrides the SA applied to the whole peer group.

### Topology Diagram

Figure 55 on page 457 shows the topology used in this example.

Figure 55: IPsec for BGP



### 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.

```
[edit]
set security ipsec security-association test-sa mode transport
set security ipsec security-association test-sa manual direction bidirectional protocol
  esp
set security ipsec security-association test-sa manual direction bidirectional spi 1000
set security ipsec security-association test-sa manual direction bidirectional encryption
  algorithm 3des-cbc
set security ipsec security-association test-sa manual direction bidirectional encryption
  key ascii-text
  "$9$KPT3AtOIhr6/u1lhvM8X7Vb2JGimfz.PtuB1hcs2goGDkqf5Qndb.5QzCA0BIrvx7VsgJ"
set protocols bgp group 1 neighbor 1.1.1.1 ipsec-sa test-sa
```

**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 Router R1:

1. Configure the SA mode.  

```
[edit security ipsec security-association test-sa]
user@R1# set mode transport
```
2. Configure the IPsec protocol to be used.  

```
[edit security ipsec security-association test-sa]
user@R1# set manual direction bidirectional protocol esp
```
3. Configure to security parameter index to uniquely identify the SA.  

```
[edit security ipsec security-association test-sa]
user@R1# set manual direction bidirectional spi 1000
```
4. Configure the encryption algorithm.  

```
[edit security ipsec security-association test-sa]
user@R1# set manual direction bidirectional encryption algorithm 3des-cbc
```
5. Configure the encryption key.  

```
[edit security ipsec security-association test-sa]
user@R1# set manual direction bidirectional encryption key ascii-text
"$9$KPT3At01hr6/u1lhvM8X7Vb2JGimfz.PtuB1hcs2goGDkqf5Qndb.5QzCA0BIrVx7VsgJ"
```

When you use an ASCII text key, the key must contain exactly 24 characters.
6. Apply the SA to the BGP peer.  

```
[edit protocols bgp group 1 neighbor 1.1.1.1]
user@R1# set ipsec-sa test-sa
```

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

```
user@R1# show protocols
bgp {
  group 1 {
    neighbor 1.1.1.1 {
      ipsec-sa test-sa;
    }
  }
}

user@R1# show security
ipsec {
  security-association test-sa {
    mode transport;
    manual {
      direction bidirectional {
        protocol esp;
        spi 1000;
      }
    }
  }
}
```

```
        encryption {
            algorithm 3des-cbc;
            key ascii-text
                "$9$kPT3AtO1hr6/u1lhvM8X7Vb2JGimfzPtuBilhcs2goGDkqf5Qndb.5QzCA0BIRvx7VsgJ!";
            ## SECRET-DATA
        }
    }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on Router R0, changing only the neighbor address.

Verification

Confirm that the configuration is working properly.

- [Verifying the Security Associaton on page 459](#)

Verifying the Security Associaton

<b>Purpose</b>	Make sure that the correct settings appear in the output of the <b>show ipsec security-associations</b> command.																		
<b>Action</b>	From operational mode, enter the <b>show ipsec security-associations</b> command.  user@R1> <b>show ipsec security-associations</b> Security association: test-sa <table><tr><th>Direction</th><th>SPI</th><th>AUX-SPI</th><th>Mode</th><th>Type</th><th>Protocol</th></tr><tr><td>inbound</td><td>1000</td><td>0</td><td>transport</td><td>manual</td><td>ESP</td></tr><tr><td>outbound</td><td>1000</td><td>0</td><td>transport</td><td>manual</td><td>ESP</td></tr></table>	Direction	SPI	AUX-SPI	Mode	Type	Protocol	inbound	1000	0	transport	manual	ESP	outbound	1000	0	transport	manual	ESP
Direction	SPI	AUX-SPI	Mode	Type	Protocol														
inbound	1000	0	transport	manual	ESP														
outbound	1000	0	transport	manual	ESP														
<b>Meaning</b>	The output is straightforward for most fields except the AUX-SPI field. The AUX-SPI is the value of the auxiliary security parameter index. When the value is AH or ESP, AUX-SPI is always 0. When the value is AH+ESP, AUX-SPI is always a positive integer.																		
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Configuring Manual IPsec Security Associations for an ES PIC</i></li></ul>																		

Examples: Configuring TCP and BGP Security

- [Understanding Security Options for BGP with TCP on page 460](#)
- [Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers on page 460](#)
- [Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 465](#)
- [Example: Limiting TCP Segment Size for BGP on page 468](#)

## Understanding Security Options for BGP with TCP

Among routing protocols, BGP is unique in using TCP as its transport protocol. BGP peers are established by manual configuration between routing devices to create a TCP session on port 179. A BGP-enabled device periodically sends keepalive messages to maintain the connection.

Over time, BGP has become the dominant interdomain routing protocol on the Internet. However, it has limited guarantees of stability and security. Configuring security options for BGP must balance suitable security measures with acceptable costs. No one method has emerged as superior to other methods. Each network administrator must configure security measures that meet the needs of the network being used.

For detailed information about the security issues associated with BGP's use of TCP as a transport protocol, see RFC 4272, *BGP Security Vulnerabilities Analysis*.

### Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers

This example shows how to configure a standard stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except from specified BGP peers.

- [Requirements on page 460](#)
- [Overview on page 460](#)
- [Configuration on page 461](#)
- [Verification on page 464](#)

---

#### Requirements

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

---

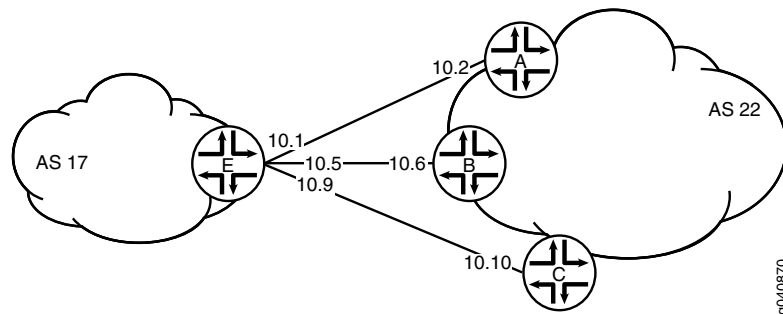
#### Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except the specified BGP peers.

The stateless firewall filter **filter\_bgp179** matches all packets from the directly connected interfaces on Device A and Device B to the destination port number 179.

[Figure 56 on page 461](#) shows the topology used in this example. Device C attempts to make a TCP connection to Device E. Device E blocks the connection attempt. This example shows the configuration on Device E.

Figure 56: Typical Network with BGP Peer Sessions



### 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 C**

```

set interfaces ge-1/2/0 unit 10 description to-E
set interfaces ge-1/2/0 unit 10 family inet address 10.10.10.10/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 17
set protocols bgp group external-peers neighbor 10.10.10.9
set routing-options autonomous-system 22

```

**Device E**

```

set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-1/2/1 unit 5 description to-B
set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-1/0/0 unit 9 description to-C
set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 2 family inet filter input filter_bgp179
set interfaces lo0 unit 2 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set routing-options autonomous-system 17
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.2/32
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.6/32
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then accept
set firewall family inet filter filter_bgp179 term 2 then reject

```

### Configuring Device E

**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 Device E with a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requestors except specified BGP peers:

1. Configure the interfaces.

```
user@E# set interfaces ge-1/2/0 unit 0 description to-A
user@E# set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
```

```
user@E# set interfaces ge-1/2/1 unit 5 description to-B
user@E# set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30
```

```
user@E# set interfaces ge-1/0/0 unit 9 description to-C
user@E# set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
```

2. Configure BGP.

```
[edit protocols bgp group external-peers]
user@E# set type external
user@E# set peer-as 22
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10
```

3. Configure the autonomous system number.

```
[edit routing-options]
user@E# set autonomous-system 17
```

4. Define the filter term that accepts TCP connection attempts to port 179 from the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@E# set term 1 from source-address 10.10.10.2/32
user@E# set term 1 from source-address 10.10.10.6/32
user@E# set term 1 from destination-port bgp
user@E# set term 1 then accept
```

5. Define the other filter term to reject packets from other sources.

```
[edit firewall family inet filter filter_bgp179]
user@E# set term 2 then reject
```

6. Apply the firewall filter to the loopback interface.

```
[edit interfaces lo0 unit 2 family inet]
user@E# set filter input filter_bgp179
user@E# set address 192.168.0.1/32
```

**Results** From configuration mode, confirm your configuration by entering the **show firewall**, **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not

display the intended configuration, repeat the instructions in this example to correct the configuration.

```

user@E# show firewall
family inet {
  filter filter_bgp179 {
    term 1 {
      from {
        source-address {
          10.10.10.2/32;
          10.10.10.6/32;
        }
        destination-port bgp;
      }
      then accept;
    }
    term 2 {
      then {
        reject;
      }
    }
  }
}

user@E# show interfaces
lo0 {
  unit 2 {
    family inet {
      filter {
        input filter_bgp179;
      }
      address 192.168.0.1/32;
    }
  }
}
ge-1/2/0 {
  unit 0 {
    description to-A;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
ge-1/2/1 {
  unit 5 {
    description to-B;
    family inet {
      address 10.10.10.5/30;
    }
  }
}
ge-1/0/0 {
  unit 9 {
    description to-C;
    family inet {
      address 10.10.10.9/30;
    }
  }
}

```

```

    }
  }
}

user@E# show protocols
bgp {
  group external-peers {
    type external;
    peer-as 22;
    neighbor 10.10.10.2;
    neighbor 10.10.10.6;
    neighbor 10.10.10.10;
  }
}

user@E# show routing-options
autonomous-system 17;

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying That the Filter Is Configured on page 464](#)
- [Verifying the TCP Connections on page 464](#)
- [Monitoring Traffic on the Interfaces on page 465](#)

### *Verifying That the Filter Is Configured*

**Purpose** Make sure that the filter is listed in output of the **show firewall filter** command.

**Action** user@E> show firewall filter filter\_bgp179  
Filter: filter\_bgp179

### *Verifying the TCP Connections*

**Purpose** Verify the TCP connections.

**Action** From operational mode, run the **show system connections extensive** command on Device C and Device E.

The output on Device C shows the attempt to establish a TCP connection. The output on Device E shows that connections are established with Device A and Device B only.

user@C> show system connections extensive | match 10.10.10

```
tcp4      0      0  10.10.10.9.51872      10.10.10.10.179      SYN_SENT
```

user@E> show system connections extensive | match 10.10.10

```
tcp4      0      0  10.10.10.5.179        10.10.10.6.62096     ESTABLISHED
tcp4      0      0  10.10.10.6.62096      10.10.10.5.179       ESTABLISHED
tcp4      0      0  10.10.10.1.179        10.10.10.2.61506     ESTABLISHED
tcp4      0      0  10.10.10.2.61506      10.10.10.1.179       ESTABLISHED
```

### Monitoring Traffic on the Interfaces

**Purpose** Use the `monitor traffic` command to compare the traffic on an interface that establishes a TCP connection with the traffic on an interface that does not establish a TCP connection.

**Action** From operational mode, run the `monitor traffic` command on the Device E interface to Device B and on the Device E interface to Device C. The following sample output verifies that in the first example, acknowledgment (**ack**) messages are received. In the second example, **ack** messages are not received.

```
user@E> monitor traffic size 1500 interface ge-1/2/1.5
19:02:49.700912 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P
3330573561:3330573580(19) ack 915601686 win 16384 <nop,nop,timestamp 1869518816
1869504850>: BGP, length: 19
19:02:49.801244 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 19 win 16384
<nop,nop,timestamp 1869518916 1869518816>
19:03:03.323018 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: P 1:20(19) ack 19 win
16384 <nop,nop,timestamp 1869532439 1869518816>: BGP, length: 19
19:03:03.422418 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: . ack 20 win 16384
<nop,nop,timestamp 1869532539 1869532439>
19:03:17.220162 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P 19:38(19) ack 20 win
16384 <nop,nop,timestamp 1869546338 1869532439>: BGP, length: 19
19:03:17.320501 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 38 win 16384
<nop,nop,timestamp 1869546438 1869546338>
```

```
user@E> monitor traffic size 1500 interface ge-1/0/0.9
```

```
18:54:20.175471 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869009240 0,sackOK,eol>
18:54:23.174422 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869012240 0,sackOK,eol>
18:54:26.374118 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869015440 0,sackOK,eol>
18:54:29.573799 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:32.773493 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:35.973185 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
```

### Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List

This example shows how to configure a standard stateless firewall filter that limits certain TCP and Internet Control Message Protocol (ICMP) traffic destined for the Routing Engine by specifying a list of prefix sources that contain allowed BGP peers.

- [Requirements on page 465](#)
- [Overview on page 466](#)
- [Configuration on page 466](#)
- [Verification on page 468](#)

#### Requirements

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

## Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except BGP peers that have a specified prefix.

A source prefix list, **plist\_bgp179**, is created that specifies the list of source prefixes that contain allowed BGP peers.

The stateless firewall filter **filter\_bgp179** matches all packets from the source prefix list **plist\_bgp179** to the destination port number 179.

## Configuration

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*.

- [Configure the Filter on page 466](#)
- [Results on page 467](#)

### 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.

```
set policy-options prefix-list plist_bgp179 apply-path "protocols bgp group <*> neighbor <*>"
set firewall family inet filter filter_bgp179 term 1 from source-address 0.0.0.0/0
set firewall family inet filter filter_bgp179 term 1 from source-prefix-list plist_bgp179 except
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then reject
set firewall family inet filter filter_bgp179 term 2 then accept
set interfaces lo0 unit 0 family inet filter input filter_bgp179
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

### Configure the Filter

### 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 filter:

1. Expand the prefix list **bgp179** to include all prefixes pointed to by the BGP peer group defined by **protocols bgp group <\*> neighbor <\*>**.

```
[edit policy-options prefix-list plist_bgp179]
user@host# set apply-path "protocols bgp group <*> neighbor <*>"
```

2. Define the filter term that rejects TCP connection attempts to port 179 from all requesters except the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term1 from source-address 0.0.0.0/0
user@host# set term term1 from source-prefix-list bgp179 except
```

```

user@host# set term term1 from destination-port bgp
user@host# set term term1 then reject

```

3. Define the other filter term to accept all packets.

```

[edit firewall family inet filter filter_bgp179]
user@host# set term term2 then accept

```

4. Apply the firewall filter to the loopback interface.

```

[edit interfaces lo0 unit 0 family inet]
user@host# set filter input filter_bgp179
user@host# set address 127.0.0.1/32

```

### Results

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

```

user@host# show firewall
family inet {
  filter filter_bgp179 {
    term 1 {
      from {
        source-address {
          0.0.0.0/0;
        }
        source-prefix-list {
          plist_bgp179 except;
        }
        destination-port bgp;
      }
      then {
        reject;
      }
    }
    term 2 {
      then {
        accept;
      }
    }
  }
}

user@host# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input filter_bgp179;
      }
      address 127.0.0.1/32;
    }
  }
}

```

```

user@host# show policy-options
prefix-list plist_bgp179 {
    apply-path "protocols bgp group <*> neighbor <*>";
}

```

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

Repeat the procedure, where appropriate, for every BGP-enabled device in the network, using the appropriate interface names and addresses for each BGP-enabled device.

### Verification

Confirm that the configuration is working properly.

#### *Displaying the Firewall Filter Applied to the Loopback Interface*

- Purpose** Verify that the firewall filter **filter\_bgp179** is applied to the IPv4 input traffic at logical interface **lo0.0**.
- Action** Use the **show interfaces statistics** operational mode command for logical interface **lo0.0**, and include the **detail** option. Under the **Protocol inet** section of the command output section, the **Input Filters** field displays the name of the stateless firewall filter applied to the logical interface in the input direction:

```

[edit]
user@host> show interfaces statistics lo0.0 detail
Logical interface lo0.0 (Index 321) (SNMP ifIndex 16) (Generation 130)
  Flags: SNMP-Traps Encapsulation: Unspecified
  Traffic statistics:
    Input bytes : 0
    Output bytes : 0
    Input packets: 0
    Output packets: 0
  Local statistics:
    Input bytes : 0
    Output bytes : 0
    Input packets: 0
    Output packets: 0
  Transit statistics:
    Input bytes : 0 0 bps
    Output bytes : 0 0 bps
    Input packets: 0 0 pps
    Output packets: 0 0 pps
  Protocol inet, MTU: Unlimited, Generation: 145, Route table: 0
    Flags: Sendbroadcast-pkt-to-re
    Input Filters: filter_bgp179
    Addresses, Flags: Primary
      Destination: Unspecified, Local: 127.0.0.1, Broadcast: Unspecified,
      Generation: 138

```

### Example: Limiting TCP Segment Size for BGP

This example shows how to avoid Internet Control Message Protocol (ICMP) vulnerability issues by limiting TCP segment size when you are using maximum transmission unit

(MTU) discovery. Using MTU discovery on TCP paths is one method of avoiding BGP packet fragmentation.

- [Requirements on page 469](#)
- [Overview on page 469](#)
- [Configuration on page 470](#)
- [Verification on page 472](#)
- [Troubleshooting on page 472](#)

### Requirements

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

### Overview

TCP negotiates a maximum segment size (MSS) value during session connection establishment between two peers. The MSS value negotiated is primarily based on the maximum transmission unit (MTU) of the interfaces to which the communicating peers are directly connected. However, due to variations in link MTU on the path taken by the TCP packets, some packets in the network that are well within the MSS value might be fragmented when the packet size exceeds the link's MTU.

To configure the TCP MSS value, include the `tcp-mss` statement with a segment size from 1 through 4096.

If the router receives a TCP packet with the SYN bit and the MSS option set, and the MSS option specified in the packet is larger than the MSS value specified by the `tcp-mss` statement, the router replaces the MSS value in the packet with the lower value specified by the `tcp-mss` statement.

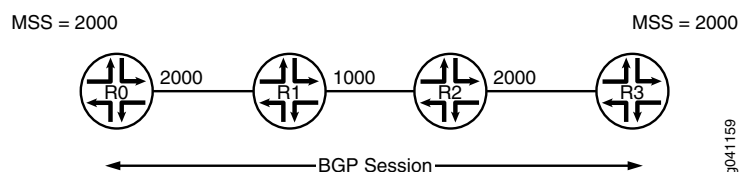
The configured MSS value is used as the maximum segment size for the sender. The assumption is that the TCP MSS value used by the sender to communicate with the BGP neighbor is the same as the TCP MSS value that the sender can accept from the BGP neighbor. If the MSS value from the BGP neighbor is less than the MSS value configured, the MSS value from the BGP neighbor is used as the maximum segment size for the sender.

This feature is supported with TCP over IPv4 and TCP over IPv6.

### Topology Diagram

[Figure 57 on page 469](#) shows the topology used in this example.

**Figure 57: TCP Maximum Segment Size for BGP**



## 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.

```
R0
set interfaces fe-1/2/0 unit 1 family inet address 1.1.0.1/30
set interfaces lo0 unit 1 family inet address 10.255.14.179/32
set protocols bgp group-int tcp-mss 2020
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.179
set protocols bgp group int mtu-discovery
set protocols bgp group int neighbor 10.255.71.24 tcp-mss 2000
set protocols bgp group int neighbor 10.255.14.177
set protocols bgp group int neighbor 10.0.14.4 tcp-mss 4000
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface 10.255.14.179
set routing-options autonomous-system 65000
```

**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 Router R0:

1. Configure the interfaces.  

```
[edit interfaces]
user@R0# set fe-1/2/0 unit 1 family inet address 1.1.0.1/30
user@R0# set lo0 unit 1 family inet address 10.255.14.179/32
```
2. Configure an interior gateway protocol (IGP), OSPF in this example.  

```
[edit protocols ospf area 0.0.0.0]
user@R0# set interface fe-1/2/0.1
user@R0# set interface 10.255.14.179
```
3. Configure one or more BGP groups.  

```
[edit protocols bgp group int]
user@R0# set type internal
user@R0# set local-address 10.255.14.179
```
4. Configure MTU discovery to prevent packet fragmentation.  

```
[edit protocols bgp group int]
user@R0# set mtu-discovery
```
5. Configure the BGP neighbors, with the TCP MSS set globally for the group or specifically for the various neighbors.  

```
[edit protocols bgo group int]
user@R0# set tcp-mss 2020
user@R0# set neighbor 10.255.14.177
user@R0# set neighbor 10.255.71.24 tcp-mss 2000
user@R0# set neighbor 10.0.14.4 tcp-mss 4000
```



**NOTE:** The TCP MSS neighbor setting overrides the group setting.

6. Configure the local autonomous system.

```
[edit routing-options]
user@R0# set autonomous-system 65000
```

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

```
user@R0# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 1.1.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.255.14.179/32;
    }
  }
}

user@R0# show protocols
bgp {
  group int {
    type internal;
    local-address 10.255.14.179;
    mtu-discovery;
    tcp-mss 2020;
    neighbor 10.255.71.24 {
      tcp-mss 2000;
    }
    neighbor 10.255.14.177;
    neighbor 10.0.14.4 {
      tcp-mss 4000;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.1;
    interface 10.255.14.179;
  }
}

user@R0# show routing-options
autonomous-system 65000;
```

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

## Verification

To confirm that the configuration is working properly, run the following commands:

- **show system connections extensive | find <neighbor-address>**, to check the negotiated TCP MSS value.
- **monitor traffic interface**, to monitor BGP traffic and to make sure that the configured TCP MSS value is used as the MSS option in the TCP SYN packet.

## Troubleshooting

- [MSS Calculation with MTU Discovery on page 472](#)

### MSS Calculation with MTU Discovery

**Problem** Consider an example in which two routing devices (R1 and R2) have an internal BGP (IBGP) connection. On both of the routers, the connected interfaces have 4034 as the IPv4 MTU.

```
user@R1# show protocols bgp | display set
[edit]
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 45.45.45.2
set protocols bgp group ibgp mtu-discovery
set protocols bgp group ibgp neighbor 45.45.45.1
```

```
user@R1# run show interfaces xe-0/0/3 extensive | match mtu
```

```
Link-level type: Ethernet, MTU: 4048, LAN-PHY mode, Speed: 10Gbps,
FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Protocol inet, MTU: 4034, Generation: 180, Route table: 0
Protocol multiservice, MTU: Unlimited, Generation: 181, Route table: 0
```

In the following packet capture on Device R1, the negotiated MSS is 3994. In the **show system connections extensive** information for MSS, it is set to 2048.

```
05:50:01.575218 Out
  Juniper PCAP Flags [Ext], PCAP Extension(s) total length 16
    Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
    Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
    Device Interface Index Extension TLV #1, length 2, value: 137
    Logical Interface Index Extension TLV #4, length 4, value: 69
  -----original packet-----
  00:21:59:e1:e8:03 > 00:19:e2:20:79:01, ethertype IPv4 (0x0800), length
78: (tos 0xc0, ttl 64, id 53193, offset 0, flags [DF], proto: TCP (6), length:
64) 45.45.45.2.62840 > 45.45.45.1.bgp: S 2939345813:2939345813(0) win 16384 **mss
3994,nop,wscale 0,nop,nop,timestamp 70559970 0,sackOK,eol>
05:50:01.575875 In
  Juniper PCAP Flags [Ext, no-L2, In], PCAP Extension(s) total length 16
    Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
    Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
    Device Interface Index Extension TLV #1, length 2, value: 137
    Logical Interface Index Extension TLV #4, length 4, value: 69
```

```

-----original packet-----
PFE proto 2 (ipv4): (tos 0xc0, ttl 255, id 37709, offset 0, flags [DF], proto:
TCP (6), length: 64) 45.45.45.1.bgp > 45.45.45.2.62840: S 2634967984:2634967984(0)
ack 2939345814 win 16384 **mss 3994,nop,wscale 0,nop,nop,timestamp 174167273
70559970,sackOK,eol>

```

```
user@R1# run show system connections extensive | find 45.45
```

```

tcp4          0      0 45.45.45.2.62840          45.45.45.1.179
                ESTABLISHED
  sndsbcc:          0 sndsbmbcnt:          0 sndsbmbmax:      131072
sndsblowat:      2048 sndsbhiwat:      16384
  rcvsbcc:          0 rcvsbmbcnt:          0 rcvsbmbmax:      131072
rcvsblowat:          1 rcvsbhiwat:      16384
  proc id:      19725  proc name:      rpd
    iss: 2939345813    sndup: 2939345972
  snduna: 2939345991    sndnxt: 2939345991    sndwnd:      16384
  sndmax: 2939345991    sndcwnd:      10240 sndssthresh: 1073725440
    irs: 2634967984    rcvup: 2634968162
  rcvnxt: 2634968162    rcvadv: 2634984546    rcvwnd:      16384
    rtt:          0    srtt:      1538    rttv:      1040
  rxtcur:      1200    rxtshift:          0    rtseq: 2939345972
  rttmin:      1000  mss:      2048

```

**Solution** This is expected behavior with Junos OS. The MSS value is equal to the MTU value minus the IP or IPv6 and TCP headers. This means that the MSS value is generally 40 bytes less than the MTU (for IPv4) and 60 bytes less than the MTU (for IPv6). This value is negotiated between the peers. In this example, it is  $4034 - 40 = 3994$ . Junos OS then rounds this value to a multiple of 2 KB. The value is  $3994 / 2048 * 2048 = 2048$ . So it is not necessary to see same MSS value with in the **show system connections** output.

$3994 / 2048 = 1.95$

1.95 is rounded to 1.

$1 * 2048 = 2048$

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Configuring Origin Validation for BGP

- [Use Case and Benefit of Origin Validation on page 473](#)
- [Understanding Origin Validation for BGP on page 474](#)
- [Example: Configuring Origin Validation for BGP on page 480](#)

### Use Case and Benefit of Origin Validation

If an administrator of an autonomous system (AS) begins advertising all or part of another company's assigned network, BGP has no built-in method to recognize the error and respond in a way that would avoid service interruptions.

Suppose, for example, that an administrator in a customer network mistakenly advertises a route (let's say 10.65.153.0/24) directing traffic to the customer's service provider AS 1. This /24 route is a more specific route than the one used by the actual content provider (10.65.152.0/22) which directs traffic to AS 2. Because of the way routers work, most routers select the more specific route and send traffic to AS 1 instead of AS 2.

The hijacked prefix is seen widely across the Internet as transit routers propagate the updated path information. The invalid routes can be distributed broadly across the Internet as the routers in the default free zone (DFZ) carry the hijacked route. Eventually the correct AS path is restored to BGP peers, but in the meantime service interruptions are to be expected.

Because BGP relies on a transitive trust model, validation between customer and provider is important. In the example above, the service provider AS 1 did not validate the faulty advertisement for 10.65.153.0/24. By accepting this advertisement and readvertising it to its peers and providers, AS 1 was propagating the wrong route. The routers that received this route from AS 1 selected it because it was a more specific route. The actual content provider was advertising 10.65.152.0/22 before the mistake occurred. The /24 was a smaller (and more specific) advertisement. According to the usual BGP route selection process, the /24 was then chosen, effectively completing the hijack.

Even with fast detection and reaction of the content provider and cooperation with other providers, service for their prefix can be interrupted for many minutes up to several hours. The exact duration of the outage depends on your vantage point on the Internet. When these sorts of events occur, there is renewed interest in solutions to this vulnerability. BGP is fundamental to provider relationships and will not be going away anytime soon. This example demonstrates a solution that uses origin validation. This solution relies on cryptographic extensions to BGP and a distributed client-server model that avoids overtaxing router CPUs.

Origin validation helps to overcome the vulnerability of transitive trust by enabling a provider to limit the advertisements it accepts from a customer. The mechanics involve the communication of routing policies based on an extended BGP community attribute.

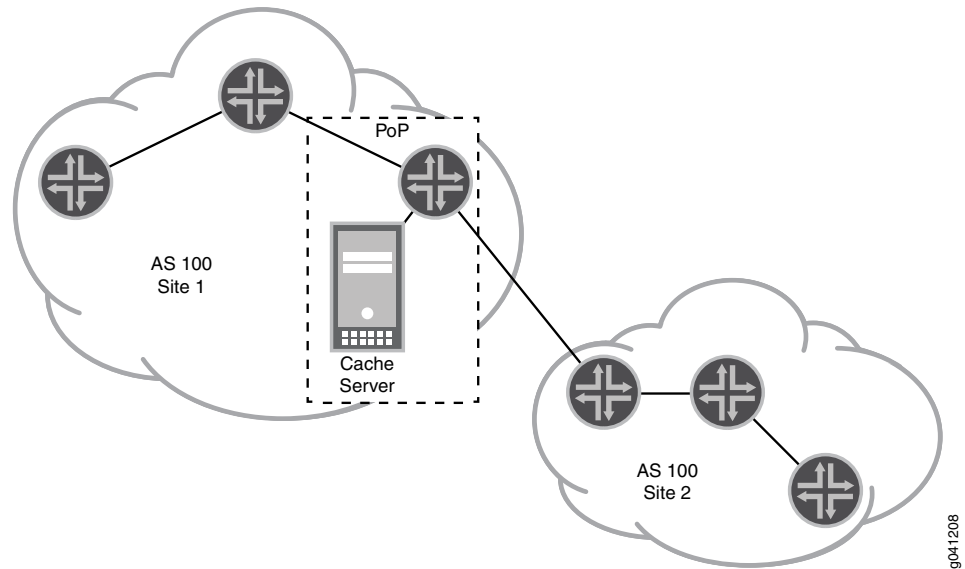
## Understanding Origin Validation for BGP

Origin validation helps to prevent the unintentional advertisement of routes. Sometimes network administrators mistakenly advertise routes to networks that they do not control. You can resolve this security issue by configuring origin validation (also known as secure interdomain routing). Origin validation is a mechanism by which route advertisements can be authenticated as originating from an expected autonomous system (AS). Origin validation uses one or more resource public key infrastructure (RPKI) cache servers to perform authentication for specified BGP prefixes. To authenticate a prefix, the router (BGP speaker) queries the database of validated prefix-to-AS mappings, which are downloaded from the cache server, and ensures that the prefix originated from an expected AS.

Junos OS supports origin validation for IPv4 and IPv6 prefixes.

[Figure 58 on page 475](#) shows a sample topology.

Figure 58: Sample Topology for Origin Validation



### Supported Standards

The Junos OS implementation of origin validation supports the following RFCs and draft:

- RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*
- RFC 6811, *BGP Prefix Origin Validation*
- Internet draft draft-ietf-sidr-origin-validation-signaling-00, *BGP Prefix Origin Validation State Extended Community* (partial support)

The extended community (origin validation state) is supported in Junos OS routing policy. The specified change in the route selection procedure is not supported.

### How Origin Validation Works

The RPKI and origin validation use X.509 certificates with extensions specified in RFC 3779, *X.509 Extensions for IP Addresses and AS Identifiers*.

The RPKI consists of a distributed collection of information. Each Certification Authority publishes its end-entity (EE) certificates, certificate revocation lists (CRLs), and signed objects at a particular location. All of these repositories form a complete set of information that is available to every RPKI cache server.

Each RPKI cache server maintains a local cache of the entire distributed repository collection by regularly synchronizing each element in the local cache against the original repository publication point.

On the router, the database entries are formatted as route validation (RV) records. An RV record is a (prefix, maximum length, origin AS) triple. It matches any route whose prefix matches the RV prefix, whose prefix length does not exceed the maximum length given in the RV record, and whose origin AS equals the origin AS given in the RV record.

An RV record is a simplified version of a route origin authorization (ROA). An ROA is a digitally signed object that provides a means of verifying that an IP address block holder has authorized an AS to originate routes to one or more prefixes within the address block. ROAs are not directly used in route validation. The cache server exports a simplified version of the ROA to the router as an RV record.

The maximum length value must be greater than or equal to the length of the authorized prefix and less than or equal to the length (in bits) of an IP address in the address family (32 for IPv4 and 128 for IPv6). The maximum length defines the IP address prefix that the AS is authorized to advertise.

For example, if the IP address prefix is 200.4.66/24, and the maximum length is 26, the AS is authorized to advertise 200.4.66.0/24, 200.4.66.0/25, 200.4.66.128/25, 200.4.66.0/26, 200.4.66.64/26, 200.4.66.128/26, and 200.4.66.192/26. When the maximum length is not present, the AS is only authorized to advertise exactly the prefix specified in the RV.

As another example, an RV can contain the prefix 200.4.66/24 with a maximum length of 26, as well as the prefix 200.4.66.0/28 with a maximum length of 28. This RV would authorize the AS to advertise any prefix beginning with 200.4.66 with a length of at least 24 and no greater than 26, as well as the specific prefix 200.4.66.0/28.

The origin of a route is represented by the right-most AS number in the AS\_PATH attribute. Origin validation operates by comparing the origin AS in a routing update with the authorized source AS published in RV records.

The security provided by origin validation alone is known to be weak against a determined attacker because there is no protection against such an attacker spoofing the source AS. That said, origin validation provides useful protection against accidental announcements.

Although origin validation could be implemented by having each router directly participate in the RPKI, this is seen as too resource intensive (because many public-key cryptography operations are required to validate the RPKI data) as well as operationally intensive to set up and maintain an RPKI configuration on each router. For this reason, a separate RPKI cache server performs public-key validations, and generates a validated database of prefix-to-AS mappings. The validated database is downloaded to a client router over a secure TCP connection. The router thus requires little information about the RPKI infrastructure and has no public-key cryptography requirements, other than the encrypted transport password. The router subsequently uses the downloaded data to validate received route updates.

When you configure server sessions, you can group the sessions together and configure session parameters for each session in the group. The router tries periodically to set up a configurable maximum number of connections to cache servers. If connection setup fails, a new connection attempt is made periodically.

In the meantime, after the validation import policy is applied to the BGP session, route-validation is performed irrespective of cache session state (up or down) and RV database (empty or not empty). If the RV database is empty or none of the cache server sessions are up, the validation state for each route is set to unknown, because no RV record exists to evaluate a received BGP prefix.

The retry-attempt period is configurable. After successfully connecting to a cache server, the router queries for the latest database serial number and requests that the RPKI cache transmits all of the RV entries belonging to that version of the database.

Each inbound message resets a liveliness timer for the RPKI cache server. After all updates are learned, the router performs periodic liveliness checks based on a configurable interval. This is done by sending a serial query protocol data unit (PDU) with the same serial number that the cache server reported in its latest notification PDU. The cache server responds with zero or more updates and an end-of-data (EOD) PDU, which also refreshes the liveliness state of the cache server and resets a record-lifetime timer.

When a prefix is received from an external BGP (EBGP) peer, it is examined by an import policy and marked as Valid, Invalid, Unknown, or Unverified:

- Valid—Indicates that the prefix and AS pair are found in the database.
- 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.
- Unverified—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers.

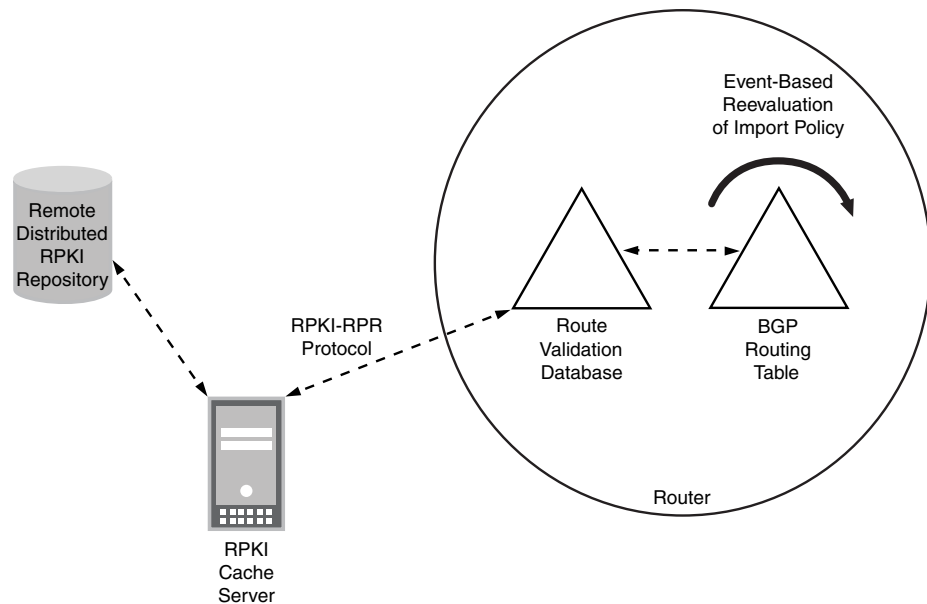
If there are any potential matches for the route in the validation database, the route has to match one of them to be valid. Otherwise, it is invalid. Any match is adequate to make the route valid. It does not need to be a best match. Only if there are no potential matches is the route considered to be unknown. For more information about the prefix-to-AS mapping database logic, see Section 2 of Internet draft draft-ietf-sidr-pfx-validate-01, *BGP Prefix Origin Validation*.

### BGP Interaction with the Route Validation Database

The route validation (RV) database contains a collection of RV records that the router downloads from the RPKI cache server. After the RV database is populated with RV records, the RV database scans the RIB-Local routing table to determine if there are any prefixes in RIB-Local that might be affected by the RV records in the database. (RIB-Local contains the IPv4 and IPv6 routes shown in the output of the **show route protocol bgp** command.)

This process triggers a BGP reevaluation of BGP import policies (not export policies).

Figure 59 on page 478 shows the process.

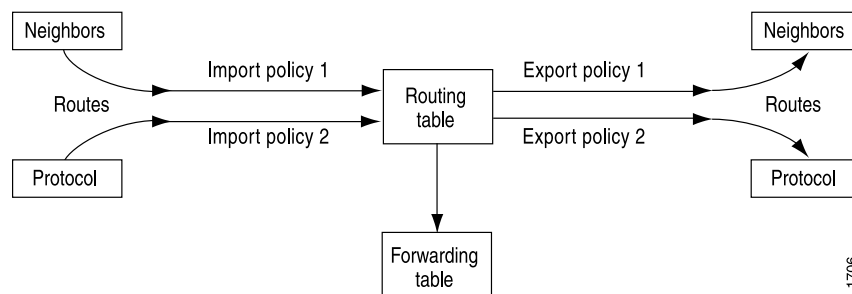


g041209

Import policies are applied to RIB-In. Another way to understand this is that Import policies are applied to the routes that are shown in the output of the **show route receive-protocol bgp** command, while export policies are applied to routes that are shown by the **show route advertising-protocol bgp** command.

As shown in Figure 60 on page 478, you use import routing policies to control which routes BGP places in the routing table, and export routing policies to control which routes BGP advertises from the routing table to its neighbors.

**Figure 60: Importing and Exporting Routing Policies**



1706

When you configure a route-validation import policy, the policy configuration uses a **validation-database** match condition. This match condition triggers a query in the RV database for the validation state of a prefix in a given routing instance. The default operation is to query the validation database matching the routing instance. If no route validation instance is found, the master instance is queried.

In the following BGP import policy, the **from validation-database** condition triggers a lookup in the router's RV database. An action is taken if the validation state is valid. The action is to accept the route and set the **validation-state** in the routing table to valid.

```

[edit protocols bgp]
import validation;

[edit policy-options]
policy-statement validation-1 {
  term valid {
    from {
      protocol bgp;
      validation-database valid; # Triggers a lookup in the RV database
    }
    then {
      validation-state valid; # Sets the validation state in the routing table
      accept;
    }
  }
}

```

### Community Attribute to Announce RPKI Validation State to IBGP Neighbors

Prefix validation is done only for external BGP (EBGP) updates. Within an AS, you likely do not want to have an RPKI session running on every internal BGP (IBGP) router. Instead, you need a way to carry the validation state across the IBGP mesh so that all IBGP speakers have consistent information. This is accomplished by carrying the validation state in a non-transitive extended community. The community attribute announces and receives the validation state of a prefix between IBGP neighbors.

Junos OS supports the following well-known extended communities for route validation:

- origin-validation-state-valid
- origin-validation-state-invalid
- origin-validation-state-unknown

The following sample BGP import policy is configured on the router that has a session with an RPKI server.

<b>Router With RPKI Session</b>	<pre> policy-statement validation-1 {   term valid {     from {       protocol bgp;       validation-database valid;     }     then {       validation-state valid;       community add origin-validation-state-valid;       accept;     }   } } </pre>
---------------------------------	---

The following sample BGP import policy is configured on an IBGP peer router that does not have a session with an RPKI server.

<b>IBGP Peer Router Without RPKI Session</b>	<pre> policy-statement validation-2 {   term valid {     from community origin-validation-state-valid; </pre>
--	---

```
        then validation-state valid;
    }
}
```

---

### Nonstop Active Routing and Origin Validation

When you configure origin validation on a router that has dual Routing Engines and nonstop active routing is enabled, both the master and the standby Routing Engines have a copy of the RV database. These two RV databases remain synchronized with each other.

The router does not maintain two identical sessions with the RPKI server. The RPKI-RTR protocol runs on the master Routing Engine only. On the standby Routing Engine, the RPKI cache server session is always down.

The RV database is actively maintained by the master Routing Engine through its session with the RPKI server. This database is replicated on the standby Routing Engine. Though the session is down on the standby Routing Engine, the replicated RV database does contain RV records. When the standby Routing Engine switches over and becomes the master Routing Engine, it already has a fully populated RV database.

To view the contents of the two databases, use the [show validation database](#) and [show validation replication database](#) commands.

---

### Marking a Prefix Range as Never Allowed

The route validation model has one major shortcoming: It only provides positive updates. It can declare which AS is the legitimate owner of a prefix. However, it cannot explicitly convey a negative update, as in: This prefix is never originated by a given AS. This functionality can be provided to some extent using an AS 0 workaround.

The Junos OS implementation does not attempt to restrict its inputs from the cache. For example, an RV record with origin AS 0 is installed and matched upon just like any other. This enables a workaround to mark a prefix range as never allowed to be announced because AS 0 is not a valid AS. The AS in the RV record never matches the AS received from the EBGP peer. Thus, any matching prefix is marked invalid.

## Example: Configuring Origin Validation for BGP

This example shows how to configure origin validation between BGP peers by ensuring that received route advertisements are sent (originated) from the expected autonomous system (AS). If the origin AS is validated, a policy can specify that the prefixes are, in turn, advertised.

- [Requirements on page 481](#)
- [Overview on page 481](#)
- [Configuration on page 482](#)
- [Verification on page 492](#)

## Requirements

This example has the following hardware and software requirements:

- Resource public key infrastructure (RPKI) cache server, using third-party software to authenticate BGP prefixes.
- Junos OS Release 12.2 or later running on the routing device that communicates with the cache server over a TCP connection.

## Overview

Sometimes routes are unintentionally advertised due to operator error. To prevent this security issue, you can configure BGP to validate the originating AS. This feature uses a cache server to authenticate prefixes or prefix ranges.

The following configuration statements enable origin AS validation:

```
[edit routing-options]
validation {
  group group-name {
    max-sessions number;
    session address {
      hold-time seconds;
      local-address local-ip-address;
      port port-number;
      preference number;
      record-lifetime seconds;
      refresh-time seconds;
    }
  }
  static {
    record destination {
      maximum-length prefix-length {
        origin-autonomous-system as-number {
          validation-state (invalid | valid);
        }
      }
    }
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag;
  }
}
```

This example uses default settings for the validation parameters.

Most of the available configuration statements are optional. The required settings are as follows:

```
validation {
  group group-name {
    session address {
    }
  }
}
```

```

    }
  }

```

The `[edit routing-options validation static]` hierarchy level enables you to configure static records on a routing device, thus overwriting records received from an RPKI cache server.

For example:

```

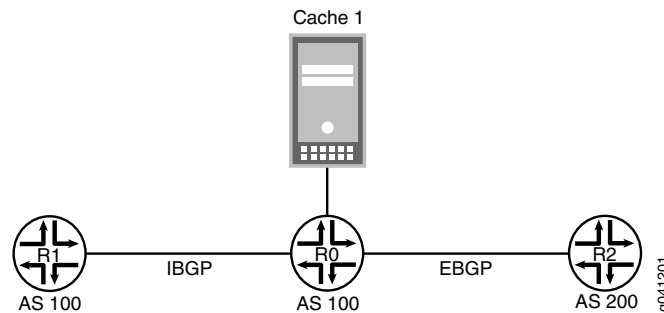
[edit routing-options validation]
user@R0# set static record 10.0.0.0/16 maximum-length 24 origin-autonomous-system
200 validation-state valid

```

You can configure a routing policy that operates based on the validation state of a route prefix. You can use a community attribute to announce and receive the validation state of a prefix between external BGP (EBGP) and internal BGP (IBGP) peers. Using a routing policy might be more convenient on some routers than configuring a session with an RPKI server. This example demonstrates the use of the validation-state community attribute between IBGP peers.

Figure 61 on page 482 shows the sample topology.

**Figure 61: Topology for Origin Validation**



In this example, Device R0 has an IBGP connection to Device R1 and an EBGP connection to Device R2. Device R0 receives route validation (RV) records from the cache server using the protocol defined in Internet draft draft-ietf-sidr-rpki-rtr-19, *The RPKI/Router Protocol* to send the RV records. The RPKI-Router Protocol runs over TCP. The RV records are used by Device R0 to build a local RV database. On Device R1, the validation state is set based on the BGP community called validation-state, which is received with the route.

### Configuration

- [Configuring Device R0 on page 484](#)
- [Configuring Device R1 on page 488](#)
- [Configuring Device R2 on page 490](#)

#### 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 R0
set interfaces ge-1/2/0 unit 2 description to-R1
set interfaces ge-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces ge-1/2/1 unit 5 description to-R2

```

```

set interfaces ge-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces ge-1/2/2 unit 9 description to-cache
set interfaces ge-1/2/2 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 1 family inet address 1.0.1.1/32
set protocols bgp group int type internal
set protocols bgp group int local-address 1.0.1.1
set protocols bgp group int export send-direct
set protocols bgp group int neighbor 1.1.1.1
set protocols bgp group ext type external
set protocols bgp group ext import validation
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.6
set protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set policy-options policy-statement send-direct from protocol direct
set policy-options policy-statement send-direct then accept
set policy-options policy-statement validation term valid from protocol bgp
set policy-options policy-statement validation term valid from validation-database valid
set policy-options policy-statement validation term valid then local-preference 110
set policy-options policy-statement validation term valid then validation-state valid
set policy-options policy-statement validation term valid then community add
    origin-validation-state-valid
set policy-options policy-statement validation term valid then accept
set policy-options policy-statement validation term invalid from protocol bgp
set policy-options policy-statement validation term invalid from validation-database
    invalid
set policy-options policy-statement validation term invalid then local-preference 90
set policy-options policy-statement validation term invalid then validation-state invalid
set policy-options policy-statement validation term invalid then community add
    origin-validation-state-invalid
set policy-options policy-statement validation term invalid then accept
set policy-options policy-statement validation term unknown from protocol bgp
set policy-options policy-statement validation term unknown then validation-state
    unknown
set policy-options policy-statement validation term unknown then community add
    origin-validation-state-unknown
set policy-options policy-statement validation term unknown then accept
set policy-options community origin-validation-state-invalid members 0x43:100:2
set policy-options community origin-validation-state-unknown members 0x43:100:1
set policy-options community origin-validation-state-valid members 0x43:100:0
set routing-options autonomous-system 100
set routing-options validation group test session 10.0.0.10

```

```

Device R1  set interfaces ge-1/2/0 unit 1 family inet address 10.0.0.1/30
            set interfaces lo0 unit 2 family inet address 1.1.1.1/32
            set protocols bgp group int type internal
            set protocols bgp group int local-address 1.1.1.1
            set protocols bgp group int import validation-ibgp
            set protocols bgp group int neighbor 1.0.1.1
            set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
            set protocols ospf area 0.0.0.0 interface lo0.2 passive
            set policy-options policy-statement validation-ibgp term valid from community
                origin-validation-state-valid
            set policy-options policy-statement validation-ibgp term valid then validation-state valid

```

```

set policy-options policy-statement validation-ibgp term invalid from community
  origin-validation-state-invalid
set policy-options policy-statement validation-ibgp term invalid then validation-state
  invalid
set policy-options policy-statement validation-ibgp term unknown from community
  origin-validation-state-unknown
set policy-options policy-statement validation-ibgp term unknown then validation-state
  unknown
set policy-options community origin-validation-state-invalid members 0x43:100:2
set policy-options community origin-validation-state-unknown members 0x43:100:1
set policy-options community origin-validation-state-valid members 0x43:100:0
set routing-options autonomous-system 100

```

**Device R2**

```

set interfaces ge-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 5 family inet address 172.16.1.1/32
set interfaces lo0 unit 5 family inet address 192.168.2.3/32
set interfaces lo0 unit 5 family inet address 2.2.0.2/32
set protocols bgp group ext export send-direct
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 10.0.0.5
set policy-options policy-statement send-direct from protocol direct
set policy-options policy-statement send-direct from protocol local
set policy-options policy-statement send-direct then accept
set routing-options autonomous-system 200

```

### Configuring Device R0

**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 Device R0:

1. Configure the interfaces.

```

[edit interfaces]
user@R0# set ge-1/2/0 unit 2 description to-R1
user@R0# set ge-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R0# set ge-1/2/1 unit 5 description to-R2
user@R0# set ge-1/2/1 unit 5 family inet address 10.0.0.5/30

user@R0# set ge-1/2/2 unit 9 description to-cache
user@R0# set ge-1/2/2 unit 9 family inet address 10.0.0.9/30

user@R0# set lo0 unit 1 family inet address 1.0.1.1/32

```

2. Configure BGP.

Apply the **send-direct** export policy so that direct routes are exported from the routing table into BGP.

Apply the **validation** import policy to set the validation-state and BGP community attributes for all the routes imported (or received) from Device R0's EBGP peers.

Configure an IBGP session with Device R1. Configure an EBGP session with Device R2.

```
[edit protocols bgp]
user@R0# set group int type internal
user@R0# set group int local-address 1.0.1.1
user@R0# set group int export send-direct
user@R0# set group int neighbor 1.1.1.1
```

```
user@R0# set group ext type external
user@R0# set group ext import validation
user@R0# set group ext export send-direct
user@R0# set group ext peer-as 200
user@R0# set group ext neighbor 10.0.0.6
```

3. Configure OSPF (or another interior gateway protocol [IGP]) on the interface that faces the IBGP peer and on the loopback interface.



**NOTE:** If you use the loopback interface address in the IBGP neighbor statement, you must enable an IGP on the loopback interface. Otherwise, the IBGP session is not established.

```
[edit protocols ospf area 0.0.0.0]
user@R0# set interface ge-1/2/0.2
user@R0# set interface lo0.1 passive
```

4. Configure the routing policy that exports direct routes from the routing table into BGP.

```
[edit policy-options policy-statement send-direct]
user@R0# set from protocol direct
user@R0# set then accept
```

5. Configure the routing policy that specifies attributes to be modified based on the validation state of each BGP route.

```
[edit policy-options policy-statement validation]
user@R0# set term valid from protocol bgp
user@R0# set term valid from validation-database valid
user@R0# set term valid then local-preference 110
user@R0# set term valid then validation-state valid
user@R0# set term valid then community add origin-validation-state-valid
user@R0# set term valid then accept
```

```
user@R0# set term invalid from protocol bgp
user@R0# set term invalid from validation-database invalid
user@R0# set term invalid then local-preference 90
user@R0# set term invalid then validation-state invalid
user@R0# set term invalid then community add origin-validation-state-invalid
user@R0# set term invalid then accept
```

```
user@R0# set term unknown from protocol bgp
user@R0# set term unknown then validation-state unknown
```

```

user@R0# set term unknown then community add origin-validation-state-unknown
user@R0# set term unknown then accept

```

```

[edit policy-options]
user@R0# set community origin-validation-state-invalid members 0x43:100:2
user@R0# set community origin-validation-state-unknown members 0x43:100:1
user@R0# set community origin-validation-state-valid members 0x43:100:0

```

6. Configure the session with the RPKI cache server.

```

[edit routing-options validation]
user@R0# set group test session 10.0.0.10

```

7. Configure the autonomous system (AS) number.

```

[edit routing-options]
user@R0# set autonomous-system 100

```

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

```

user@R0# show interfaces
ge-1/2/0 {
  unit 2 {
    description to-R1;
    family inet {
      address 10.0.0.2/30;
    }
  }
}
ge-1/2/1 {
  unit 5 {
    description to-R2;
    family inet {
      address 10.0.0.5/30;
    }
  }
}
ge-1/2/2 {
  unit 9 {
    description to-cache;
    family inet {
      address 10.0.0.9/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 1.0.1.1/32;
    }
  }
}

```

```
user@R0# show protocols
bgp {
  group int {
    type internal;
    local-address 1.0.1.1;
    export send-direct;
    neighbor 1.1.1.1;
  }
  group ext {
    type external;
    import validation;
    export send-direct;
    peer-as 200;
    neighbor 10.0.0.6;
  }
}
ospf {
  area 0.0.0.0 {
    interface ge-1/2/0.2;
    interface lo0.1 {
      passive;
    }
  }
}

user@R0# show policy-options
policy-statement send-direct {
  from protocol direct;
  then accept;
}
policy-statement validation {
  term valid {
    from {
      protocol bgp;
      validation-database valid;
    }
    then {
      local-preference 110;
      validation-state valid;
      community add origin-validation-state-valid;
      accept;
    }
  }
  term invalid {
    from {
      protocol bgp;
      validation-database invalid;
    }
    then {
      local-preference 90;
      validation-state invalid;
      community add origin-validation-state-invalid;
      accept;
    }
  }
  term unknown {
```

```

        from protocol bgp;
        then {
            validation-state unknown;
            community add origin-validation-state-unknown;
            accept;
        }
    }
}
community origin-validation-state-invalid members 0x43:100:2;
community origin-validation-state-unknown members 0x43:100:1;
community origin-validation-state-valid members 0x43:100:0;

user@R0# show routing-options
autonomous-system 100;
validation {
    group test {
        session 10.0.0.10;
    }
}

```

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

### Configuring Device R1

**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 Device R1:

1. Configure the interfaces.

```

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

user@R1# set lo0 unit 2 family inet address 1.1.1.1/32

```

2. Configure BGP.

Apply the **validation-ibgp** import policy to set the validation-state and BGP community attributes for all the routes received from Device R1's IBGP peers.

Configure an IBGP session with Device R0.

```

[edit protocols bgp group int]
user@R1# set type internal
user@R1# set local-address 1.1.1.1
user@R1# set import validation-ibgp
user@R1# set neighbor 1.0.1.1

```

3. Configure OSPF.

```

[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-1/2/0.1
user@R1# set interface lo0.2 passive

```

4. Configure the routing policy that specifies attributes to be modified based on the validation-state BGP community attribute of the BGP routes received from Device RO.

```
[edit policy-options policy-statement validation-ibgp]
user@R1# set term valid from community origin-validation-state-valid
user@R1# set term valid then validation-state valid

user@R1# set term invalid from community origin-validation-state-invalid
user@R1# set term invalid then validation-state invalid

user@R1# set term unknown from community origin-validation-state-unknown
user@R1# set term unknown then validation-state unknown

[edit policy-options]
user@R1# set community origin-validation-state-invalid members 0x43:100:2
user@R1# set community origin-validation-state-unknown members 0x43:100:1
user@R1# set community origin-validation-state-valid members 0x43:100:0
```

5. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 100
```

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

```
user@R1# show interfaces
ge-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}

user@R1# show protocols
bgp {
  group int {
    type internal;
    local-address 1.1.1.1;
    import validation-ibgp;
    neighbor 1.0.1.1;
  }
}
ospf {
```

```
area 0.0.0.0 {
  interface ge-1/2/0.1;
  interface lo0.2 {
    passive;
  }
}

user@R1# show policy-options
policy-statement validation-ibgp {
  term valid {
    from community origin-validation-state-valid;
    then validation-state valid;
  }
  term invalid {
    from community origin-validation-state-invalid;
    then validation-state invalid;
  }
  term unknown {
    from community origin-validation-state-unknown;
    then validation-state unknown;
  }
}
community origin-validation-state-invalid members 0x43:100:2;
community origin-validation-state-unknown members 0x43:100:1;
community origin-validation-state-valid members 0x43:100:0;
}

user@R1# show routing-options
autonomous-system 100;
```

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

### **Configuring Device R2**

**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 Device R2:

1. Configure the interfaces.

Several addresses are configured on the loopback interface to serve as routes for demonstration purposes.

[edit interfaces]

```
user@R2# set ge-1/2/0 unit 6 family inet address 10.0.0.6/30
```

```
user@R2# set lo0 unit 5 family inet address 172.16.1.1/32
```

```
user@R2# set lo0 unit 5 family inet address 192.168.2.3/32
```

```
user@R2# set lo0 unit 5 family inet address 2.2.0.2/32
```

2. Configure BGP.

[edit protocols bgp]

```
user@R2# set group ext export send-direct
```

```

user@R2# set group ext peer-as 100
user@R2# set group ext neighbor 10.0.0.5

```

3. Configure the routing policy.

```

[edit policy-options policy-statement send-direct]
user@R2# set from protocol direct
user@R2# set from protocol local
user@R2# set then accept

```

4. Configure the autonomous system (AS) number.

```

[edit routing-options]
user@R2# set autonomous-system 200

```

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

```

user@R2# show interfaces
ge-1/2/0 {
  unit 6 {
    family inet {
      address 10.0.0.6/30;
    }
  }
}
lo0 {
  unit 5 {
    family inet {
      address 172.16.1.1/32;
      address 192.168.2.3/32;
      address 2.2.0.2/32;
    }
  }
}

user@R2# show protocols
bgp {
  group ext {
    export send-direct;
    peer-as 100;
    neighbor 10.0.0.5;
  }
}

user@R2# show policy-options
policy-statement send-direct {
  from protocol [ direct local ];
  then accept;
}

user@R2# show routing-options
autonomous-system 200;

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying That the Modified Attributes Are Displayed in the Routing Tables on page 492](#)
- [Using Trace Operations on page 493](#)
- [Displaying Validation Information on page 493](#)

### *Verifying That the Modified Attributes Are Displayed in the Routing Tables*

**Purpose** Verify that the BGP routes on Device R0 and Device R1 have the expected validation states and the expected local preferences.

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

```

user@R0> show route
inet.0: 12 destinations, 13 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.0.1.1/32      * [Direct/0] 04:53:39
                 > via lo0.1
1.1.1.1/32      * [OSPF/10] 04:50:53, metric 1
                 > to 10.0.0.1 via lt-1/2/0.2
2.2.0.2/32      * [BGP/170] 01:30:37, localpref 110
                 AS path: 200 I, validation-state: valid
                 > to 10.0.0.6 via lt-1/2/0.5
10.0.0.0/30     * [Direct/0] 04:51:44
                 > via lt-1/2/0.2
10.0.0.2/32     * [Local/0] 04:51:45
                 Local via lt-1/2/0.2
10.0.0.4/30     * [Direct/0] 04:51:44
                 > via lt-1/2/0.5
                 [BGP/170] 02:24:57, localpref 110
                 AS path: 200 I, validation-state: valid
                 > to 10.0.0.6 via lt-1/2/0.5
10.0.0.5/32     * [Local/0] 04:51:45
                 Local via lt-1/2/0.5
10.0.0.8/30     * [Direct/0] 03:01:28
                 > via lt-1/2/0.9
10.0.0.9/32     * [Local/0] 04:51:45
                 Local via lt-1/2/0.9
172.16.1.1/32   * [BGP/170] 02:24:57, localpref 90
                 AS path: 200 I, validation-state: invalid
                 > to 10.0.0.6 via lt-1/2/0.5
192.168.2.3/32  * [BGP/170] 02:24:57, localpref 100
                 AS path: 200 I, validation-state: validation-state: unknown
                 > to 10.0.0.6 via lt-1/2/0.5
224.0.0.5/32    * [OSPF/10] 04:53:46, metric 1
                 MultiRecv

user@R1> show route
inet.0: 10 destinations, 12 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.0.2/32      * [BGP/170] 01:06:58, localpref 110, from 1.0.1.1
                 AS path: 200 I, validation-state: valid
                 > to 10.0.0.2 via lt-1/2/0.1
172.16.1.1/32   * [BGP/170] 00:40:52, localpref 90, from 1.0.1.1

```

```

AS path: 200 I, validation-state: invalid
> to 10.0.0.2 via 1t-1/2/0.1
192.168.2.3/32 * [BGP/170] 01:06:58, localpref 100, from 1.0.1.1
AS path: 200 I, validation-state: unknown
> to 10.0.0.2 via 1t-1/2/0.1
224.0.0.5/32 * [OSPF/10] 04:57:09, metric 1
MultiRecv

```

**Meaning** The routes have the expected validation states and local-preference values, based on information received from the RPKI cache server.

### *Using Trace Operations*

**Purpose** Configure trace operations for origin validation, and monitor the results of a newly advertised route.

**Action** • On Device R0, configure tracing.

```

[edit routing-options validation traceoptions]
user@R0# set file rv-tracing
user@R0# set flag all

```

```

user@R0# commit

```

• On Device R2, add a route by adding another address on the loopback interface.

```

[edit interfaces lo0 unit 5 family inet]
user@R2# set address 10.4.4.4/32

```

```

user@R2# commit

```

• On Device R0, check the trace file.

```

user@R0> file show /var/log/rv-tracing
Jan 27 11:27:43.804803 rv_get_policy_state: rt 10.4.4.4/32 origin-as 200,
validation result valid
Jan 27 11:27:43.944037 task_job_create_background: create prio 7 job
Route-validation GC for task Route Validation
Jan 27 11:27:43.986580 background dispatch running job Route-validation GC for
task Route Validation
Jan 27 11:27:43.987374 task_job_delete: delete background job Route-validation
GC for task Route Validation
Jan 27 11:27:43.987463 background dispatch completed job Route-validation GC
for task Route Validation

```

**Meaning** Route validation is operating as expected.

### *Displaying Validation Information*

**Purpose** Run the various validation commands.

**Action** user@R0> [show validation statistics](#)

```
Total RV records: 3
Total Replication RV records: 3
  Prefix entries: 3
  Origin-AS entries: 3
Memory utilization: 9789 bytes
Policy origin-validation requests: 114
  Valid: 32
  Invalid: 54
  Unknown: 28
BGP import policy reevaluation notifications: 156
  inet.0, 156
  inet6.0, 0
```

user@R0> [show validation database](#)

RV database for instance master

Prefix	Origin-AS	Session	State
Mismatch			
2.0.0.0/8-32	200	10.0.0.10	valid
10.0.0.0/8-32	200	10.0.0.10	valid
172.0.0.0/8-12	200	10.0.0.10	invalid

```
IPv4 records: 3
IPv6 records: 0
```

user@R0> [show validation replication database](#)

RRV replication database for instance master

Prefix	Origin-AS	Session	State
2.0.0.0/8-32	200	10.0.0.10	valid
10.0.0.0/8-32	200	10.0.0.10	valid
172.0.0.0/8-12	200	10.0.0.10	invalid

```
IPv4 records: 3
IPv6 records: 0
```

user@R0> [show validation group](#)

master

```
Group: test, Maximum sessions: 2
Session 10.0.0.10, State: Connect, Preference: 100
```

user@R0> [show validation session](#)

Session	State	Flaps	Uptime	#IPv4/IPv6
records				
10.0.0.10	Up	0	00:02:28	1/0

user@R0> [request validation policy](#)

```
Enqueued 3 IPv4 records
Enqueued 0 IPv6 records
```

## Related Documentation

- [Understanding BGP Communities and Extended Communities as Routing Policy Match Conditions on page 223](#)
- [How BGP Communities and Extended Communities Are Evaluated in Routing Policy Match Conditions](#)

- *Example: Configuring Extended Communities in a Routing Policy*



## CHAPTER 11

# BGP Flap Configuration

- [Example: Preventing BGP Session Resets on page 497](#)
- [Examples: Configuring BGP Flap Damping on page 505](#)
- [Example: Configuring Error Handling for BGP Update Messages on page 525](#)

### Example: Preventing BGP Session Resets

---

- [Understanding BGP Session Resets on page 497](#)
- [Example: Preventing BGP Session Flaps When VPN Families Are Configured on page 497](#)

### Understanding BGP Session Resets

Certain configuration actions and events cause BGP sessions to be reset (dropped and then reestablished).

If you configure both route reflection and VPNs on the same routing device, the following modifications to the route reflection configuration cause current BGP sessions to be reset:

- Adding a cluster ID—If a BGP session shares the same autonomous system (AS) number with the group where you add the cluster ID, all BGP sessions are reset regardless of whether the BGP sessions are contained in the same group.
- Creating a new route reflector—If you have an internal BGP (IBGP) group with an AS number and create a new route reflector group with the same AS number, all BGP sessions in the IBGP group and the new route reflector group are reset.
- Changing configuration statements that affect BGP peers, such as renaming a BGP group, resets the BGP sessions.
- If you change the address family specified in the **[edit protocols bgp family]** hierarchy level, all current BGP sessions on the routing device are dropped and then reestablished.

### Example: Preventing BGP Session Flaps When VPN Families Are Configured

This example shows a workaround for a known issue in which BGP sessions sometimes go down and then come back up (in other words, flap) when virtual private network (VPN) families are configured. If any VPN family (for example, **inet-vpn**, **inet6-vpn**, **inet-mpvn**, **inet-mdt**, **inet6-mpvn**, **l2vpn**, **iso-vpn**, and so on) is configured on a BGP master instance, a flap of either a route reflector (RR) internal BGP (IBGP) session or an external

BGP (EBGP) session causes flaps of other BGP sessions configured with the same VPN family.

- [Requirements on page 498](#)
- [Overview on page 499](#)
- [Configuration on page 501](#)
- [Verification on page 504](#)

## Requirements

---

Before you begin:

- Configure router interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure VPNs.

## Overview

---

When a router or switch is configured as either a route reflector (RR) or an AS boundary router (an external BGP peer) and a VPN family (for example, the **family inet-vpn unicast** statement) is configured, a flap of either the RR IBGP session or the EBGp session causes flaps of all other BGP sessions that are configured with the **family inet-vpn unicast** statement. This example shows how to prevent these unnecessary session flaps.

The reason for the flapping behavior is related to BGP operation in Junos OS when originating VPN routes.

BGP has the following two modes of operation with respect to originating VPN routes:

- If BGP does not need to propagate VPN routes because the session has no EBGp peer and no RR clients, BGP exports VPN routes directly from the *instance.inet.0* routing table to other PE routers. This behavior is efficient in that it avoids the creation of two copies of many routes (one in the *instance.inet.0* table and one in the *bgp.l3vpn.0* table).
- If BGP does need to propagate VPN routes because the session has an EBGp peer or RR clients, BGP first exports the VPN routes from the *instance.inet.0* table to the *bgp.l3vpn.0* table. Then BGP exports the routes to other PE routers. In this scenario, two copies of the route are needed to enable best-route selection. A PE router might receive the same VPN route from a CE device and also from an RR client or EBGp peer.



.....

**NOTE:** The route export is not performed if the route in *instance.inet.0* is a secondary route. In Junos OS, a route is only exported one time from one routing table as a primary route to another routing table as a secondary route. Because the route in *instance.inet.0* is already a secondary route, it is not allowed to be moved again to the *bgp.l3vpn.0* table, as needed to be advertised. The route does not reach the *bgp.l3vpn.0* table and thus is not advertised. One workaround is to send the routes that should be advertised to *inet.0* so that they are advertised.

.....

When, because of a configuration change, BGP transitions from needing two copies of a route to not needing two copies of a route (or the reverse), all sessions over which VPN routes are exchanged go down and then come back up. Although this example focuses on the **family inet-vpn unicast** statement, the concept applies to all VPN network layer reachability information (NLRI) families. This issue impacts logical systems as well. All BGP sessions in the master instance related to the VPN NLRI family are brought down to implement the table advertisement change for the VPN NLRI family. Changing an RR to a non-RR or the reverse (by adding or removing the **cluster** statement) causes the table advertisement change. Also, configuring the first EBGp session or removing the EBGp session from the configuration in the master instance for a VPN NLRI family causes the table advertisement change.

The way to prevent these unnecessary session flaps is to configure an extra RR client or EBGp session as a passive session with a neighbor address that does not exist. This example focuses on the EBGp case, but the same workaround works for the RR case.

When a session is passive, the routing device does not send Open requests to a peer. Once you configure the routing device to be passive, the routing device does not originate the TCP connection. However, when the routing device receives a connection from the peer and an Open message, it replies with another BGP Open message. Each routing device declares its own capabilities.

Figure 62 on page 501 shows the topology for the EBGP case. Router R1 has an IBGP session with Routers R2 and R3 and an EBGP session with Router R4. All sessions have the **family inet-vpn unicast** statement configured. If the R1-R4 EBGP session flaps, the R1-R2 and R1-R3 BGP sessions flap also.

Figure 62: Topology for the EBGP Case

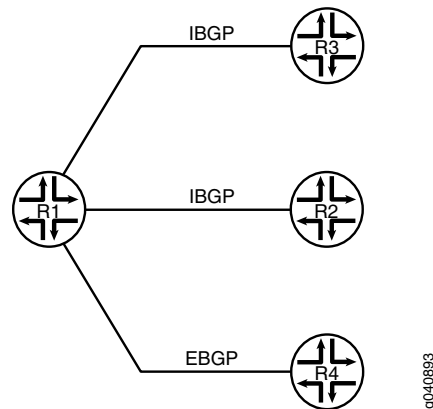
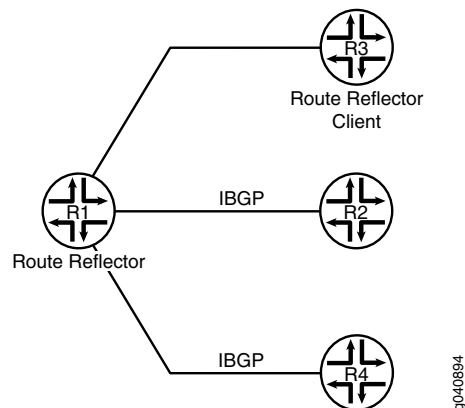


Figure 63 on page 501 shows the topology for the RR case. Router R1 is the RR, and Router R3 is the client. Router R1 has IBGP sessions with Routers R2 and R3. All sessions have the **family inet-vpn unicast** statement configured. If the R1-R3 session flaps, the R1-R2 and R1-R4 sessions flap also.

Figure 63: Topology for the RR Case



### 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.

```
set protocols bgp family inet-vpn unicast
set protocols bgp family l2vpn signaling
set protocols bgp group R1-R4 type external
set protocols bgp group R1-R4 local-address 4.4.4.2
set protocols bgp group R1-R4 neighbor 4.4.4.1 peer-as 200
set protocols bgp group R1-R2-R3 type internal
set protocols bgp group R1-R2-R3 log-updown
set protocols bgp group R1-R2-R3 local-address 15.15.15.15
set protocols bgp group R1-R2-R3 neighbor 12.12.12.12
set protocols bgp group R1-R2-R3 neighbor 13.13.13.13
set protocols bgp group Fake type external
set protocols bgp group Fake passive
set protocols bgp group Fake neighbor 100.100.100.100 peer-as 500
```

**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 EBGp scenario:

1. Configure one or more VPN families.

```
[edit protocols bgp]
user@R1# set family inet-vpn unicast
user@R1# set family l2vpn signaling
```

2. Configure the EBGp session.

```
[edit protocols bgp]
user@R1# set group R1-R4 type external
user@R1# set group R1-R4 local-address 4.4.4.2
user@R1# set group R1-R4 neighbor 4.4.4.1 peer-as 200
```

3. Configure the IBGP sessions.

```
[edit protocols bgp]
user@R1# set group R1-R2-R3 type internal
user@R1# set group R1-R2-R3 local-address 15.15.15.15
user@R1# set group R1-R2-R3 neighbor 12.12.12.12
user@R1# set group R1-R2-R3 neighbor 13.13.13.13
```

4. (Optional) Configure BGP so that it generates a **syslog** message whenever a BGP peer makes a state transition.

```
[edit protocols bgp]
user@R1# set group R1-R2-R3 log-updown
```

Enabling the **log-updown** statement causes BGP state transitions to be logged at **warning** level.

**Step-by-Step Procedure** To verify that unnecessary session flaps are occurring:

1. Run the **show bgp summary** command to verify that the sessions have been established.

```

user@R1> show bgp summary
Groups: 2 Peers: 3 Down peers: 0
Table      Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0      0      0      0      0      0      0
bgp.12vpn.0 0      0      0      0      0      0      0
inet.0      0      0      0      0      0      0      0
Peer      AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
4.4.4.1    200 6      5      0      0      1:08 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0
12.12.12.12 100 3      7      0      0      1:18 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0
13.13.13.13 100 3      6      0      0      1:14 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0

```

2. Deactivate the EBGP session.

```

user@R1# deactivate group R1-R4
user@R1# commit

```

```

Mar 10 18:27:40 R1: rpd[1464]: bgp_peer_delete:6589: NOTIFICATION sent to 4.4.4.1 (External AS 200): code
6 (Cease) subcode 3 (Peer Unconfigured), Reason: Peer Deletion
Mar 10 18:27:40 R1: rpd[1464]: bgp_adv_main_update:7253: NOTIFICATION sent to 12.12.12.12 (Internal AS
100): code 6 (Cease) subcode 6 (Other Configuration Change), Reason: Configuration change - VPN table
advertise
Mar 10 18:27:40 R1: rpd[1464]: bgp_adv_main_update:7253: NOTIFICATION sent to 13.13.13.13 (Internal AS
100): code 6 (Cease) subcode 6 (Other Configuration Change), Reason: Configuration change - VPN table
advertise

```

3. Run the **show bgp summary** command to view the session flaps.

```

user@R1> show bgp summary
Groups: 1 Peers: 2 Down peers: 2
Table      Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0      0      0      0      0      0      0
bgp.12vpn.0 0      0      0      0      0      0      0
inet.0      0      0      0      0      0      0      0
Peer      AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12 100 4      9      0      1      19 Active
13.13.13.13 100 4      8      0      1      19 Active

user@R1> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table      Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0      0      0      0      0      0      0
bgp.12vpn.0 0      0      0      0      0      0      0
inet.0      0      0      0      0      0      0      0
Peer      AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12 100 2      3      0      1      0 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0
13.13.13.13 100 2      3      0      1      0 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0

```

**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 prevent unnecessary BGP session flaps:

1. Add a passive EBGp session with a neighbor address that does not exist in the peer autonomous system (AS).

```
[edit protocols bgp]
user@R1# set group Fake type external
user@R1# set group Fake passive
user@R1# set neighbor 100.100.100.100 peer-as 500
```

2. Run the **show bgp summary** command to verify that the real sessions have been established and the passive session is idle.

```
user@R1> show bgp summary
Groups: 3 Peers: 4 Down peers: 1
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0 0 0 0 0 0
bgp.12vpn.0 0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
4.4.4.1 200 9500 9439 0 0 2d 23:14:23 Estab1
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0
12.12.12.12 100 10309 10239 0 0 3d 5:17:49 Estab1
bgp.13vpn.0: 0/0/0/0
13.13.13.13 100 10306 10241 0 0 3d 5:18:25 Estab1
bgp.13vpn.0: 0/0/0/0
100.100.100.100 500 0 0 0 0 2d 23:38:52 Idle
```

## Verification

Confirm that the configuration is working properly.

- [Bringing Down the EBGp Session on page 504](#)
- [Verifying That the IBGP Sessions Remain Up on page 504](#)

### *Bringing Down the EBGp Session*

**Purpose** Try to cause the flap issue after the workaround is configured.

**Action** user@R1# deactivate group R1-R4  
user@R1# commit

### *Verifying That the IBGP Sessions Remain Up*

**Purpose** Make sure that the IBGP sessions do not flap after the EBGp session is deactivated.

```

Action user@R1> show bgp summary
Groups: 2 Peers: 3 Down peers: 1
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0 0 0 0 0 0
bgp.12vpn.0 0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12 100 10312 10242 0 0 3d 5:19:01 Establ
bgp.13vpn.0: 0/0/0/0
13.13.13.13 100 10309 10244 0 0 3d 5:19:37 Establ
bgp.13vpn.0: 0/0/0/0
100.100.100.100 500 0 0 0 0 2d 23:40:04 Idle

user@R1> show bgp summary
Groups: 3 Peers: 4 Down peers: 1
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 0 0 0 0 0 0
bgp.12vpn.0 0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
4.4.4.1 200 5 4 0 0 28 Establ
bgp.13vpn.0: 0/0/0/0
bgp.12vpn.0: 0/0/0/0
12.12.12.12 100 10314 10244 0 0 3d 5:19:55 Establ
bgp.13vpn.0: 0/0/0/0
13.13.13.13 100 10311 10246 0 0 3d 5:20:31 Establ
bgp.13vpn.0: 0/0/0/0
100.100.100.100 500 0 0 0 0 2d 23:40:58 Idle

```

- Related Documentation**
- [Understanding External BGP Peering Sessions on page 33](#)
  - [BGP Configuration Overview on page 12](#)

## Examples: Configuring BGP Flap Damping

- [Understanding BGP Route Flap Damping Parameters on page 505](#)
- [Example: Configuring BGP Route Flap Damping Parameters on page 506](#)
- [Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family on page 515](#)

## Understanding BGP Route Flap Damping Parameters

BGP *route flapping* describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. BGP *flap damping* is a method of reducing the number of update messages sent between BGP peers, thereby reducing the load on these peers, without adversely affecting the route convergence time for stable routes.

Flap damping reduces the number of update messages by marking routes as ineligible for selection as the active or preferable route. Marking routes in this way leads to some delay, or *suppression*, in the propagation of route information, but the result is increased network stability. You typically apply flap damping to external BGP (EBGP) routes (routes in different ASs). You can also apply flap damping within a confederation, between

confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.) The exception to this rule is when flap damping is applied at the address family level, which is supported in Junos OS Release 12.2 and later. When you apply flap damping at the address family level, it works for both IBGP and EBGP.

By default, route flap damping is not enabled. Damping is applied to external peers and to peers at confederation boundaries.

When you enable damping, default parameters are applied, as summarized in [Table 6 on page 506](#).

**Table 6: Damping Parameters**

Damping Parameter	Description	Default Value	Possible Values
<b>half-life <i>minutes</i></b>	Decay half-life—Number of minutes after which an arbitrary value is halved if a route stays stable.	15 (minutes)	1 through 45
<b>max-suppress <i>minutes</i></b>	Maximum hold-down time for a route, in minutes.	60 (minutes)	1 through 720
<b>reuse</b>	Reuse threshold—Arbitrary value below which a suppressed route can be used again.	750	1 through 20,000
<b>suppress</b>	Cutoff (suppression) threshold—Arbitrary value above which a route can no longer be used or included in advertisements.	3000	1 through 20,000

To change the default BGP flap damping values, you define actions by creating a named set of damping parameters and including it in a routing policy with the damping action. For the damping routing policy to work, you also must enable BGP route flap damping.

### Example: Configuring BGP Route Flap Damping Parameters

This example shows how to configure damping parameters.

- [Requirements on page 506](#)
- [Overview on page 506](#)
- [Configuration on page 507](#)
- [Verification on page 511](#)

#### Requirements

Before you begin, configure router interfaces and configure routing protocols.

#### Overview

This example has three routing devices. Device R2 has external BGP (EBGP) connections with Device R1 and Device R3.

Device R1 and Device R3 have some static routes configured for testing purposes, and these static routes are advertised through BGP to Device R2.

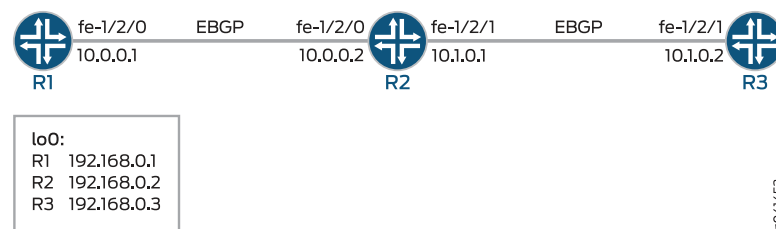
Device R2 damps routes received from Device R1 and Device R3 according to these criteria:

- Damp all prefixes with a mask length equal to or greater than 17 more aggressively than routes with a mask length between 9 and 16.
- Damp routes with a mask length between 0 and 8, inclusive, less than routes with a mask length greater than 8.
- Do not damp the 10.128.0.0/9 prefix at all.

The routing policy is evaluated when routes are being exported from the routing table into the forwarding table. Only the active routes are exported from the routing table.

Figure 64 on page 507 shows the sample network.

**Figure 64: BGP Flap Damping Topology**



“CLI Quick Configuration” on page 507 shows the configuration for all of the devices in Figure 64 on page 507.

The section “Step-by-Step Procedure” on page 508 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 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct-and-static
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct-and-static term 1 from protocol direct
set policy-options policy-statement send-direct-and-static term 1 from protocol static
set policy-options policy-statement send-direct-and-static term 1 then accept
set routing-options static route 172.16.0.0/16 reject
set routing-options static route 172.16.128.0/17 reject
set routing-options static route 172.16.192.0/20 reject
set routing-options static route 10.0.0.0/9 reject
set routing-options static route 224.0.0.0/7 reject
set routing-options static route 10.224.0.0/11 reject
set routing-options static route 0.0.0.0/0 reject
set routing-options autonomous-system 100
  
```

```

Device R2
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
  
```

```

set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp damping
set protocols bgp group ext type external
set protocols bgp group ext import damp
set protocols bgp group ext export send-direct
set protocols bgp group ext neighbor 10.0.0.1 peer-as 100
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement damp term 1 from route-filter 10.128.0.0/9 exact
damping dry
set policy-options policy-statement damp term 1 from route-filter 0.0.0.0/0
prefix-length-range /0-/8 damping timid
set policy-options policy-statement damp term 1 from route-filter 0.0.0.0/0
prefix-length-range /17-/32 damping aggressive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options damping aggressive half-life 30
set policy-options damping aggressive suppress 2500
set policy-options damping timid half-life 5
set policy-options damping dry disable
set routing-options autonomous-system 200

```

**Device R3**

```

set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct-and-static
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.1.0.1
set policy-options policy-statement send-direct-and-static term 1 from protocol direct
set policy-options policy-statement send-direct-and-static term 1 from protocol static
set policy-options policy-statement send-direct-and-static term 1 then accept
set routing-options static route 10.128.0.0/9 reject
set routing-options autonomous-system 300

```

**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 damping parameters:

1. Configure the interfaces.

```

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

user@R2# set fe-1/2/1 unit 0 family inet address 10.1.0.1/30

user@R2# set lo0 unit 0 family inet address 192.168.0.2/32

```

2. Configure the BGP neighbors.

```

[edit protocols bgp group ext]
user@R2# set type external
user@R2# set neighbor 10.0.0.1 peer-as 100
user@R2# set neighbor 10.1.0.2 peer-as 300

```

3. Create and configure the damping parameter groups.

```
[edit policy-options]
user@R2# set damping aggressive half-life 30
user@R2# set damping aggressive suppress 2500
user@R2# set damping timid half-life 5
user@R2# set damping dry disable
```

4. Configure the damping policy.

```
[edit policy-options policy-statement damp term 1]
user@R2# set from route-filter 10.128.0.0/9 exact damping dry
user@R2# set from route-filter 0.0.0.0/0 prefix-length-range /0-/8 damping timid
user@R2# set from route-filter 0.0.0.0/0 prefix-length-range /17-/32 damping
  aggressive
```

5. Enable damping for BGP.

```
[edit protocols bgp]
user@R2# set damping
```

6. Apply the policy as an import policy for the BGP neighbor.

```
[edit protocols bgp group ext]
user@R2# set import damp
```



**NOTE:** You can refer to the same routing policy one or more times in the same or different import statements.

7. Configure an export policy.

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

8. Apply the export policy.

```
[edit protocols bgp group ext]
user@R2# set export send-direct
```

9. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
```

```
    }
  }
  fe-1/2/1 {
    unit 0 {
      family inet {
        address 10.1.0.1/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 192.168.0.2/32;
      }
    }
  }
}

user@R2# show protocols
bgp {
  damping;
  group ext {
    type external;
    import damp;
    export send-direct;
    neighbor 10.0.0.1 {
      peer-as 100;
    }
    neighbor 10.1.0.2 {
      peer-as 300;
    }
  }
}

user@R2# show policy-options
policy-statement damp {
  term 1 {
    from {
      route-filter 10.128.0.0/9 exact damping dry;
      route-filter 0.0.0.0/0 prefix-length-range /0-/8 damping timid;
      route-filter 0.0.0.0/0 prefix-length-range /17-/32 damping aggressive;
    }
  }
}
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
damping aggressive {
  half-life 30;
  suppress 2500;
}
damping timid {
  half-life 5;
}
damping dry {
```

```

    disable;
}

user@R2# show routing-options
autonomous-system 200;

```

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

### Verification

Confirm that the configuration is working properly.

- [Causing Some Routes to Flap on page 511](#)
- [Checking the Route Flaps on page 511](#)
- [Verifying Route Flap Damping on page 512](#)
- [Displaying the Details of a Damped Route on page 513](#)
- [Verifying That Default Damping Parameters Are in Effect on page 513](#)
- [Filtering the Damping Information on page 514](#)

#### *Causing Some Routes to Flap*

**Purpose** To verify your route flap damping policy, some routes must flap. Having a live Internet feed almost guarantees that a certain number of route flaps will be present. If you have control over a remote system that is advertising the routes, you can modify the advertising router's policy to effect the advertisement and withdrawal of all routes or of a given prefix. In a test environment, you can cause routes to flap by clearing the BGP neighbors or by restarting the routing process on the BGP neighbors, as shown here.

**Action** From operational mode on Device R1 and Device R3, enter the **restart routing** command.



**CAUTION:** Use this command cautiously in a production network.

```
user@R1> restart routing
```

```
R1 started, pid 10474
```

```
user@R3> restart routing
```

```
R3 started, pid 10478
```

**Meaning** On Device R2, all of the routes from the neighbors are withdrawn and re-advertised.

#### *Checking the Route Flaps*

**Purpose** View the number of neighbor flaps.

**Action** From operational mode, enter the **show bgp summary** command.

```
user@R2> show bgp summary
```

```

Groups: 1 Peers: 2 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet.0
          12         1         11         0         11         0
Peer      AS      InPkt    OutPkt    OutQ    Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.1   100        10        10        0        4      2:50
0/9/0/9    0/0/0/0
10.1.0.2   300        10        10        0        4      2:53
1/3/1/2    0/0/0/0

```

**Meaning** This output was captured after the routing process was restarted on Device R2's neighbors four times.

### *Verifying Route Flap Damping*

**Purpose** Verify that routes are being hidden due to damping.

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

```
user@R2> show route damping suppressed
```

```
inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

0.0.0.0/0      [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
10.0.0.0/9     [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
10.0.0.0/30    [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
10.1.0.0/30    [BGP ] 00:00:15, localpref 100
                AS path: 300 I, validation-state: unverified
                > to 10.1.0.2 via fe-1/2/1.0
10.224.0.0/11  [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
172.16.0.0/16  [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
172.16.128.0/17 [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
172.16.192.0/20 [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
192.168.0.1/32 [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0
192.168.0.3/32 [BGP ] 00:00:15, localpref 100
                AS path: 300 I, validation-state: unverified
                > to 10.1.0.2 via fe-1/2/1.0
224.0.0.0/7    [BGP ] 00:00:12, localpref 100
                AS path: 100 I, validation-state: unverified
                > to 10.0.0.1 via fe-1/2/0.0

```

**Meaning** The output shows some routing instability. Eleven routes are hidden due to damping.

### *Displaying the Details of a Damped Route*

**Purpose** Display the details of damped routes.

**Action** From operational mode, enter the **show route damping suppressed 172.16.192.0/20 detail** command.

```
user@R2> show route damping suppressed 172.16.192.0/20 detail

inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
172.16.192.0/20 (1 entry, 0 announced)
    BGP          /-101
        Next hop type: Router, Next hop index: 758
        Address: 0x9414484
        Next-hop reference count: 9
        Source: 10.0.0.1
        Next hop: 10.0.0.1 via fe-1/2/0.0, selected
        Session Id: 0x100201
        State: <Hidden Ext>
        Local AS: 200 Peer AS: 100
        Age: 52
        Validation State: unverified
        Task: BGP_100.10.0.0.1+55922
        AS path: 100 I
        Localpref: 100
        Router ID: 192.168.0.1
        Merit (last update/now): 4278/4196
        damping-parameters: aggressive
        Last update: 00:00:52 First update: 01:01:55
        Flaps: 8
        Suppressed. Reusable in: 01:14:40
        Preference will be: 170
```

**Meaning** This output indicates that the displayed route has a mask length that is equal to or greater than /17, and confirms that it has been correctly mapped to the aggressive damping profile. You can also see the route's current (and last) figure of merit value, and when the route is expected to become active if it remains stable.

### *Verifying That Default Damping Parameters Are in Effect*

**Purpose** Locating a damped route with a /16 mask confirms that the default parameters are in effect.

**Action** From operational mode, enter the **show route damping suppressed detail | match 0/16** command.

```
user@R2> show route damping suppressed detail | match 0/16

172.16.0.0/16 (1 entry, 0 announced)

user@R2> show route damping suppressed 172.16.0.0/16 detail

inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
172.16.0.0/16 (1 entry, 0 announced)
```

```

BGP                               /-101
Next hop type: Router, Next hop index: 758
Address: 0x9414484
Next-hop reference count: 9
Source: 10.0.0.1
Next hop: 10.0.0.1 via fe-1/2/0.0, selected
Session Id: 0x100201
State: <Hidden Ext>
Local AS: 200 Peer AS: 100
Age: 1:58
Validation State: unverified
Task: BGP_100.10.0.0.1+55922
AS path: 100 I
Localpref: 100
Router ID: 192.168.0.1
Merit (last update/now): 3486/3202
Default damping parameters used
Last update: 00:01:58 First update: 01:03:01
Flaps: 8
Suppressed. Reusable in: 00:31:40
Preference will be: 170

```

**Meaning** Routes with a /16 mask are not impacted by the custom damping rules. Therefore, the default damping rules are in effect.

To repeat, the custom rules are as follows:

- Damp all prefixes with a mask length equal to or greater than 17 more aggressively than routes with a mask length between 9 and 16.
- Damp routes with a mask length between 0 and 8, inclusive, less than routes with a mask length greater than 8.
- Do not damp the 10.128.0.0/9 prefix at all.

#### *Filtering the Damping Information*

**Purpose** Use OR groupings or cascaded piping to simplify the determination of what damping profile is being used for routes with a given mask length.

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

```
user@R2> show route damping suppressed detail | match "0 announced | damp"
```

```

0.0.0.0/0 (1 entry, 0 announced)
    damping-parameters: timid
10.0.0.0/9 (1 entry, 0 announced)
    Default damping parameters used
    damping-parameters: aggressive
    damping-parameters: aggressive
10.224.0.0/11 (1 entry, 0 announced)
    Default damping parameters used
172.16.0.0/16 (1 entry, 0 announced)
    Default damping parameters used
172.16.128.0/17 (1 entry, 0 announced)
    damping-parameters: aggressive
172.16.192.0/20 (1 entry, 0 announced)
    damping-parameters: aggressive
192.168.0.1/32 (1 entry, 0 announced)

```

```

damping-parameters: aggressive
192.168.0.3/32 (1 entry, 0 announced)
damping-parameters: aggressive
224.0.0.0/7 (1 entry, 0 announced)
damping-parameters: timid

```

**Meaning** When you are satisfied that your EBGP routes are correctly associated with a damping profile, you can issue the **clear bgp damping** operational mode command to restore an active status to your damped routes, which will return your connectivity to normal operation.

### Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family

This example shows how to configure an multiprotocol BGP multicast VPN (also called Next-Generation MVPN) with BGP route flap damping.

- [Requirements on page 515](#)
- [Overview on page 515](#)
- [Configuration on page 516](#)
- [Verification on page 523](#)

#### Requirements

This example uses Junos OS Release 12.2. BGP route flap damping support for MBGP MVPN, specifically, and on an address family basis, in general, is introduced in Junos OS Release 12.2.

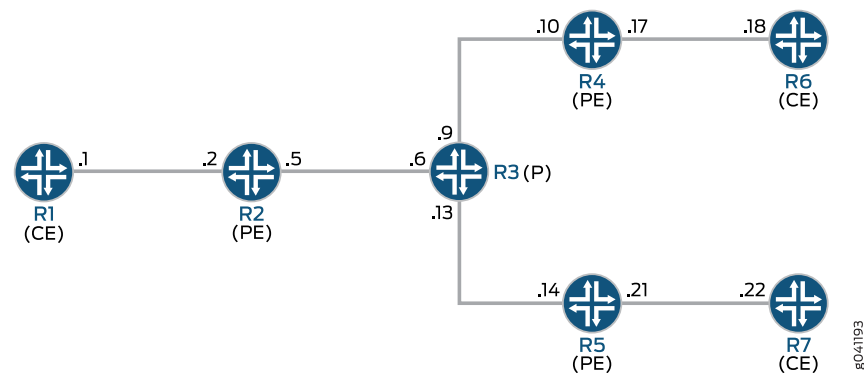
#### Overview

BGP route flap damping helps to diminish route instability caused by routes being repeatedly withdrawn and readvertised when a link is intermittently failing.

This example uses the default damping parameters and demonstrates an MBGP MVPN scenario with three provider edge (PE) routing devices, three customer edge (CE) routing devices, and one provider (P) routing device.

[Figure 65 on page 515](#) shows the topology used in this example.

**Figure 65: MBGP MVPN with BGP Route Flap Damping**



On PE Device R4, BGP route flap damping is configured for address family **inet-mvpn**. A routing policy called **dampPolicy** uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5. All other MVPN route types are not damped.

This example shows the full configuration on all devices in the “[CLI Quick Configuration](#)” on page 516 section. The “[Configuring Device R4](#)” on page 519 section shows the step-by-step configuration for PE Device R4.

### 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 ge-1/2/0 unit 1 family inet address 10.1.1.1/30
set interfaces ge-1/2/0 unit 1 family mpls
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.1

```

**Device R2**

```

set interfaces ge-1/2/0 unit 2 family inet address 10.1.1.2/30
set interfaces ge-1/2/0 unit 2 family mpls
set interfaces ge-1/2/1 unit 5 family inet address 10.1.1.5/30
set interfaces ge-1/2/1 unit 5 family mpls
set interfaces vt-1/2/0 unit 2 family inet
set interfaces lo0 unit 2 family inet address 1.1.1.2/32
set interfaces lo0 unit 102 family inet address 100.1.1.2/32
set protocols mpls interface ge-1/2/1.5
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.2
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/1.5
set protocols ldp interface ge-1/2/1.5
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/0.2
set routing-instances vpn-1 interface vt-1/2/0.2
set routing-instances vpn-1 interface lo0.102
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 provider-tunnel ldp-p2mp
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.102 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2

```

```

set routing-instances vpn-1 protocols pim interface ge-1/2/0.2 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.2
set routing-options autonomous-system 1001

```

Device R3

```

set interfaces ge-1/2/0 unit 6 family inet address 10.1.1.6/30
set interfaces ge-1/2/0 unit 6 family mpls
set interfaces ge-1/2/1 unit 9 family inet address 10.1.1.9/30
set interfaces ge-1/2/1 unit 9 family mpls
set interfaces ge-1/2/2 unit 13 family inet address 10.1.1.13/30
set interfaces ge-1/2/2 unit 13 family mpls
set interfaces lo0 unit 3 family inet address 1.1.1.3/32
set protocols mpls interface ge-1/2/0.6
set protocols mpls interface ge-1/2/1.9
set protocols mpls interface ge-1/2/2.13
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.6
set protocols ospf area 0.0.0.0 interface ge-1/2/1.9
set protocols ospf area 0.0.0.0 interface ge-1/2/2.13
set protocols ldp interface ge-1/2/0.6
set protocols ldp interface ge-1/2/1.9
set protocols ldp interface ge-1/2/2.13
set protocols ldp p2mp
set routing-options router-id 1.1.1.3

```

Device R4

```

set interfaces ge-1/2/0 unit 10 family inet address 10.1.1.10/30
set interfaces ge-1/2/0 unit 10 family mpls
set interfaces ge-1/2/1 unit 17 family inet address 10.1.1.17/30
set interfaces ge-1/2/1 unit 17 family mpls
set interfaces vt-1/2/0 unit 4 family inet
set interfaces lo0 unit 4 family inet address 1.1.1.4/32
set interfaces lo0 unit 104 family inet address 100.1.1.4/32
set protocols rsvp interface all aggregate
set protocols mpls interface all
set protocols mpls interface ge-1/2/0.10
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.4
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling damping
set protocols bgp group ibgp neighbor 1.1.1.2 import dampPolicy
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10
set protocols ldp interface ge-1/2/0.10
set protocols ldp p2mp
set policy-options policy-statement dampPolicy term term1 from family inet-mvpn
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 3
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 4
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 5
set policy-options policy-statement dampPolicy term term1 then accept
set policy-options policy-statement dampPolicy then damping no-damp
set policy-options policy-statement dampPolicy then accept

```

```

set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set policy-options damping no-damp disable
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.4
set routing-instances vpn-1 interface ge-1/2/1.17
set routing-instances vpn-1 interface lo0.104
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.104 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.17
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/1.17 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.4
set routing-options autonomous-system 1001

```

**Device R5**

```

set interfaces ge-1/2/0 unit 14 family inet address 10.1.1.14/30
set interfaces ge-1/2/0 unit 14 family mpls
set interfaces ge-1/2/1 unit 21 family inet address 10.1.1.21/30
set interfaces ge-1/2/1 unit 21 family mpls
set interfaces vt-1/2/0 unit 5 family inet
set interfaces lo0 unit 5 family inet address 1.1.1.5/32
set interfaces lo0 unit 105 family inet address 100.1.1.5/32
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.14
set protocols ldp interface ge-1/2/0.14
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.5
set routing-instances vpn-1 interface ge-1/2/1.21
set routing-instances vpn-1 interface lo0.105
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.105 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.21
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/1.21 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.5
set routing-options autonomous-system 1001

```

**Device R6**

```

set interfaces ge-1/2/0 unit 18 family inet address 10.1.1.18/30
set interfaces ge-1/2/0 unit 18 family mpls

```

```

set interfaces lo0 unit 6 family inet address 1.1.1.6/32
set protocols sap listen 224.1.1.1
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.18
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.6

```

Device R7

```

set interfaces ge-1/2/0 unit 22 family inet address 10.1.1.22/30
set interfaces ge-1/2/0 unit 22 family mpls
set interfaces lo0 unit 7 family inet address 1.1.1.7/32
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.22
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.7

```

### Configuring Device R4

**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 Device R4:

1. Configure the interfaces.

```

[edit interfaces]
user@R4# set ge-1/2/0 unit 10 family inet address 10.1.1.10/30
user@R4# set ge-1/2/0 unit 10 family mpls

user@R4# set ge-1/2/1 unit 17 family inet address 10.1.1.17/30
user@R4# set ge-1/2/1 unit 17 family mpls

user@R4# set vt-1/2/0 unit 4 family inet

user@R4# set lo0 unit 4 family inet address 1.1.1.4/32
user@R4# set lo0 unit 104 family inet address 100.1.1.4/32

```

2. Configure MPLS and the signaling protocols on the interfaces.

```

[edit protocols]
user@R4# set mpls interface all
user@R4# set mpls interface ge-1/2/0.10
user@R4# set rsvp interface all aggregate
user@R4# set ldp interface ge-1/2/0.10
user@R4# set ldp p2mp

```

3. Configure BGP.

The BGP configuration enables BGP route flap damping for the **inet-mvpn** address family. The BGP configuration also imports into the routing table the routing policy called **dampPolicy**. This policy is applied to neighbor PE Device R2.

```

[edit protocols bgp group ibgp]
user@R4# set type internal

```

```
user@R4# set local-address 1.1.1.4
user@R4# set family inet-vpn unicast
user@R4# set family inet-vpn any
user@R4# set family inet-mvpn signaling damping
user@R4# set neighbor 1.1.1.2 import dampPolicy
user@R4# set neighbor 1.1.1.5
```

4. Configure an interior gateway protocol.

```
[edit protocols ospf]
user@R4# set traffic-engineering
```

```
[edit protocols ospf area 0.0.0.0]
user@R4# set interface all
user@R4# set interface lo0.4 passive
user@R4# set interface ge-1/2/0.10
```

5. Configure a damping policy that uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5.

```
[edit policy-options policy-statement dampPolicy term term1]
user@R4# set from family inet-mvpn
user@R4# set from nlri-route-type 3
user@R4# set from nlri-route-type 4
user@R4# set from nlri-route-type 5
user@R4# set then accept
```

6. Configure the **damping** policy to disable BGP route flap damping.

The **no-damp** policy (**damping no-damp disable**) causes any damping state that is present in the routing table to be deleted. The **then damping no-damp** statement applies the **no-damp** policy as an action and has no **from** match conditions. Therefore, all routes that are not matched by **term1** are matched by this term, with the result that all other MVPN route types are not damped.

```
[edit policy-options policy-statement dampPolicy]
user@R4# set then damping no-damp
user@R4# set then accept
```

```
[edit policy-options]
user@R4# set damping no-damp disable
```

7. Configure the **parent\_vpn\_routes** to accept all other BGP routes that are not from the **inet-mvpn** address family.

This policy is applied as an OSPF export policy in the routing instance.

```
[edit policy-options policy-statement parent_vpn_routes]
user@R4# set from protocol bgp
user@R4# set then accept
```

8. Configure the VPN routing and forwarding (VRF) instance.

```
[edit routing-instances vpn-1]
user@R4# set instance-type vrf
user@R4# set interface vt-1/2/0.4
user@R4# set interface ge-1/2/1.17
user@R4# set interface lo0.104
```

```

user@R4# set route-distinguisher 100:100
user@R4# set vrf-target target:1:1
user@R4# set protocols ospf export parent_vpn_routes
user@R4# set protocols ospf area 0.0.0.0 interface lo0.104 passive
user@R4# set protocols ospf area 0.0.0.0 interface ge-1/2/1.17
user@R4# set protocols pim rp static address 100.1.1.2
user@R4# set protocols pim interface ge-1/2/1.17 mode sparse
user@R4# set protocols mvpn

```

9. Configure the router ID and the autonomous system (AS) number.

```

[edit routing-options]
user@R4# set router-id 1.1.1.4
user@R4# set autonomous-system 1001

```

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

```

user@R4# commit

```

### 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 instructions in this example to correct the configuration.

```

user@R4# show interfaces
ge-1/2/0 {
  unit 10 {
    family inet {
      address 10.1.1.10/30;
    }
    family mpls;
  }
}
ge-1/2/1 {
  unit 17 {
    family inet {
      address 10.1.1.17/30;
    }
    family mpls;
  }
}
vt-1/2/0 {
  unit 4 {
    family inet;
  }
}
lo0 {
  unit 4 {
    family inet {
      address 1.1.1.4/32;
    }
  }
}
unit 104 {
  family inet {

```

```
        address 100.1.1.4/32;
    }
}

user@R4# show protocols
rsvp {
    interface all {
        aggregate;
    }
}
mpls {
    interface all;
    interface ge-1/2/0.10;
}
bgp {
    group ibgp {
        type internal;
        local-address 1.1.1.4;
        family inet-vpn {
            unicast;
            any;
        }
        family inet-mvpn {
            signaling {
                damping;
            }
        }
        neighbor 1.1.1.2 {
            import dampPolicy;
        }
        neighbor 1.1.1.5;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface lo0.4 {
            passive;
        }
        interface ge-1/2/0.10;
    }
}
ldp {
    interface ge-1/2/0.10;
    p2mp;
}

user@R4# show policy-options
policy-statement dampPolicy {
    term term1 {
        from {
            family inet-mvpn;
            nlri-route-type [ 3 4 5 ];
        }
        then accept;
    }
}
```

```

    }
    then {
        damping no-damp;
        accept;
    }
}
policy-statement parent_vpn_routes {
    from protocol bgp;
    then accept;
}
damping no-damp {
    disable;
}

user@R4# show routing-instances
vpn-1 {
    instance-type vrf;
    interface vt-1/2/0.4;
    interface ge-1/2/1.17;
    interface lo0.104;
    route-distinguisher 100:100;
    vrf-target target:1:1;
    protocols {
        ospf {
            export parent_vpn_routes;
            area 0.0.0.0 {
                interface lo0.104 {
                    passive;
                }
                interface ge-1/2/1.17;
            }
        }
        pim {
            rp {
                static {
                    address 100.1.1.2;
                }
            }
            interface ge-1/2/1.17 {
                mode sparse;
            }
        }
        mvpn;
    }
}

user@R4# show routing-options
router-id 1.1.1.4;
autonomous-system 1001;

```

### Verification

Confirm that the configuration is working properly.

- [Verifying That Route Flap Damping Is Disabled on page 524](#)
- [Verifying Route Flap Damping on page 524](#)

**Verifying That Route Flap Damping Is Disabled**

**Purpose** Verify the presence of the **no-damp** policy, which disables damping for MVPN route types other than 3, 4, and 5.

**Action** From operational mode, enter the **show policy damping** command.

```
user@R4> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "no-damp":
  Damping disabled
```

**Meaning** The output shows that the default damping parameters are in effect and that the **no-damp** policy is also in effect for the specified route types.

**Verifying Route Flap Damping**

**Purpose** Check whether BGP routes have been damped.

**Action** From operational mode, enter the **show bgp summary** command.

```
user@R4> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0
      6      6      0      0      0      0
bgp.13vpn.2
      0      0      0      0      0      0
bgp.mvpn.0
      2      2      0      0      0      0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
1.1.1.2 1001 3159 3155 0 0 23:43:47
Establ
  bgp.13vpn.0: 3/3/3/0
  bgp.13vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
1.1.1.5 1001 3157 3154 0 0 23:43:40
Establ
  bgp.13vpn.0: 3/3/3/0
  bgp.13vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
```

**Meaning** The Damp State field shows that zero routes in the bgp.mvpn.0 routing table have been damped. Further down, the last number in the State field shows that zero routes have been damped for BGP peer 1.1.1.2.

- Related Documentation**
- [Understanding External BGP Peering Sessions on page 33](#)
  - [BGP Configuration Overview on page 12](#)

## Example: Configuring Error Handling for BGP Update Messages

- [Understanding Error Handling for BGP Update Messages on page 525](#)
- [Example: Configuring Error Handling for BGP Update Messages on page 527](#)

### Understanding Error Handling for BGP Update Messages

A BGP message is considered to be malformed when any one of the message attributes is malformed. When a router participating in a BGP session receives a malformed update message, the entire session is reset by default. This is undesirable because update messages with valid routes are also affected. To avoid this undesirable behavior, the error handling for BGP update messages needs to be modified.

To configure error handling for BGP update messages, configure the **bgp-error-tolerance** statement at the **[edit protocols bgp]**, **[edit protocols bgp group *group-name*]**, or **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level.

```
bgp-error-tolerance {
  malformed-route-limit number;
  malformed-update-log-interval seconds;
  no-malformed-route-limit;
}
```

If an attribute contains attribute flags that conflict with the value of the Attribute Type field, the attribute flags are reset to the correct value and the update message is processed. The value of the Extended Length bit in the attribute flags is unchanged because this value defines whether the attribute length is one or two octets. Hence, the value of the attribute flag affects how the BGP update packet is parsed.



**NOTE:** There is no explicit specification for the attribute flag value for the path attributes.

Malformed update messages are treated on a case by case basis, depending on the values of the attributes contained in the messages. There are three ways of handling malformed BGP update messages, listed in the decreasing order of severity.

1. **Notification message approach**—The malformed message error is logged locally, an error code update message is sent to the administration of the peer, and the entire BGP session is reset.

This approach is chosen when:

- The BGP update message contains the MP reach attribute or the MP unreachable attribute.
- The NLRI field or the BGP update message cannot be parsed correctly because of a mismatch between the attribute length and the value of the attribute length field.

2. **Treat-as-withdraw approach**—All routes within the malformed update message are treated as hidden routes, unless the **keep none** statement is configured, in which case the routes are discarded. In the absence of the **keep none** statement, the number of hidden malformed routes are configured with a limit, which when exceeded discards the routes and prevents any further malformed routes from being hidden. Junos OS removes the newly received malformed routes when the malformed route limit is reached.
3. **Attribute discard approach**—The malformed attributes in the update message are discarded; however, the message is processed. We do not recommend using this approach if the attributes to be discarded can affect route selection or installation.



**NOTE:** If an attribute appears more than once in an update message, all occurrences of the attribute, other than the first, will be discarded and the message will be processed.

The BGP update messages are scanned for the following attributes and are treated as malformed based on the values of these attributes:

- **The origin attribute**—Handled by the treat-as-withdraw approach.
- **The AS path attribute**—Handled by the treat-as-withdraw approach.
- **The AS 4 path attribute**—Handled by the attribute discard approach. If any attribute has attribute flags that conflict with the attribute type code, Junos OS resets the attribute flags to the correct value. The update message continues to be processed.

Junos OS does not change the value of the extended length bit in the attribute flags. This bit defines whether the attribute length is one octet or two octets. The value of this flag affects how the BGP packet is parsed. There is no explicit specification of this value for the path attributes.

- **The aggregator attribute**—Handled by the attribute discard approach.
- **The aggregator 4 attribute**—Handled by the attribute discard approach.
- **The next-hop attribute**—Handled by the treat-as-withdraw approach.
- **The multiple exit discriminator attribute**—Handled by the treat-as-withdraw approach.
- **The local preference attribute**—Handled by the treat-as-withdraw approach.
- **The atomic aggregate attribute**—Handled by the attribute discard approach.
- **The community attribute**—Handled by the treat-as-withdraw approach.
- **The extended community attribute**—Handled by the treat-as-withdraw approach.
- **The originator attribute**—Handled by the treat-as-withdraw approach.
- **The cluster attribute**—Handled by the treat-as-withdraw approach.
- **The PMSI attribute**—Handled by the treat-as-withdraw approach.
- **The MP reach attribute**—Handled by the notification message approach.
- **The MP unreachable attribute**—Handled by the notification message approach.

- **The attribute set attribute**—Handled by the treat-as-withdraw approach.
- **The AIGP attribute**—Handled by the treat-as-withdraw approach.
- **Unknown attribute**—If the BGP flag does not indicate that this is an optional attribute, this malformed attribute is handled by the notification message approach.



**NOTE:** When a BGP update message contains multiple malformed attributes, the most severe approach triggered by one of the attributes is followed.

## Example: Configuring Error Handling for BGP Update Messages

This example shows how to configure BGP error handling.

- [Requirements on page 527](#)
- [Overview on page 527](#)
- [Configuration on page 529](#)
- [Verification on page 532](#)

### Requirements

Before you begin:

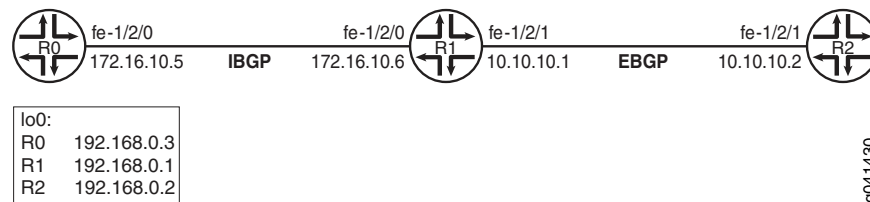
- Configure router interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure routing policies.

### Overview

When a routing device receives an update message with a malformed attribute, the router is required to reset the session. This is specified in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Session resets impact not only routes with the offending attribute, but also other valid routes exchanged over the session. Moreover, this behavior can present a potential security vulnerability in the case of optional transitive attributes. To minimize the impact on routing made by malformed update messages, the Internet draft draft-ietf-idr-error-handling-01.txt, *Revised Error Handling for BGP UPDATE Messages* specifies modifications for handling BGP update message with malformed attributes. The new error handling allows for maintaining the established session and keeping the valid routes exchanged, while removing the routes carried in the malformed UPDATE message.

In [Figure 66 on page 528](#), Device R1 has an internal BGP peering session with Device R0, and an external BGP peering session with Device R2.

Figure 66: BGP Error Handling Example Topology



To protect against malformed update messages causing network instability, Device R1 has BGP error handling configured, as shown here:

```

bgp-error-tolerance {
    malformed-update-log-interval 10;
    malformed-route-limit 5;
}

```

By default, a BGP message is considered to be malformed when any one of the message attributes is malformed. When a router participating in a BGP session receives a malformed update message, the entire session is reset. The **bgp-error-tolerance** statement overrides this behavior so that the following BGP error handling is in effect:

- For fatal errors, Junos OS sends a notification message titled Error Code Update Message and resets the BGP session. An error in the MP\_{UN}REACH attribute is considered to be fatal. The presence of multiple MP\_{UN}REACH attributes in one BGP update is also considered to be a fatal error. Junos OS resets the BGP session if it cannot parse the NLRI field or the BGP update correctly. Failure to parse the BGP update packet can happen when the attribute length does not match the length of the attribute value.
- For some nonfatal errors, Junos OS treats all the routes contained in the malformed BGP update message as withdrawn routes and installs them as hidden, unless the **keep none** statement is included in the BGP is configuration. Junos OS uses this error handling approach for the cases that involve any of the following attributes: ORIGIN, AS\_PATH, NEXT\_HOP, MULTI\_EXIT\_DISC, LOCAL\_PREF, ORIGINATOR, CLUSTER, ATTRSET, PMSI, Community, and Extended Community. In addition, if any of the mandatory well-known path attributes is missing, Junos OS treats the BGP update as malformed. To limit the memory usage of these malformed hidden routes, Junos OS stops installing new malformed hidden routes after the maximum number of such malformed hidden routes is reached. In this example, the maximum number is set to 5, using the **malformed-route-limit** statement. The default value is 1000. Optionally, you can allow an unlimited number of routes hidden due to malformed attributes. Do this by including the **no-malformed-route-limit** statement.
- For other nonfatal errors, Junos OS discards the malformed path attributes and continues to process the BGP update message. It is unsafe to use this approach on the path attributes that might affect route selection or installation. Junos OS uses this error handling approach for the cases that involve any of the following attributes: ATOMIC\_AGGREGATE, AGGREGATOR, AGGREGATOR4, and AS4PATH.

To facilitate troubleshooting of malformed packets, Junos OS logs the error listing the malformed path attribute code, flag, length, information about the peer and family, and

the first prefix from the malformed BGP update. Logging of the malformed packets might slow Junos OS performance if a significant number of malformed packets is received in a short time. To limit the performance impact, Junos OS implements an algorithm to log a malformed update, suppress logging for an interval, and log a summary. When the logging suppression timer expires, the software logs the total number of malformed attributes received during the interval. In this example, the timer is set to 10 seconds, using the **malformed-update-log-interval** statement. The default value is 300 seconds (5 minutes).

[“CLI Quick Configuration” on page 529](#) shows the configuration for all of the devices in [Figure 66 on page 528](#).

The section [“Step-by-Step Procedure” on page 530](#) describes the steps on Device R1.

### Configuration

<b>CLI Quick Configuration</b>	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 <b>[edit]</b> hierarchy level.
<b>Device R0</b>	<pre> set interfaces fe-1/2/0 unit 0 description to-R1 set interfaces fe-1/2/0 unit 0 family inet address 172.16.10.5/30 set interfaces lo0 unit 0 family inet address 192.168.0.3/32 set protocols bgp group internal-peers type internal set protocols bgp group internal-peers local-address 192.168.0.3 set protocols bgp group internal-peers export local-direct set protocols bgp group internal-peers neighbor 192.168.0.1 set protocols ospf area 0.0.0.0 interface fe-1/2/0.0 set protocols ospf area 0.0.0.0 interface lo0.0 passive set policy-options policy-statement local-direct from protocol [local direct] set policy-options policy-statement local-direct then accept set routing-options autonomous-system 64510 set routing-options router-id 192.168.0.3 </pre>
<b>Device R1</b>	<pre> set interfaces fe-1/2/1 unit 0 description to-R2 set interfaces fe-1/2/1 unit 0 family inet address 10.10.10.1/30 set interfaces fe-1/2/0 unit 0 description to-R0 set interfaces fe-1/2/0 unit 0 family inet address 172.16.10.6/30 set interfaces lo0 unit 0 family inet address 192.168.0.1/32 set protocols bgp bgp-error-tolerance malformed-update-log-interval 10 set protocols bgp bgp-error-tolerance malformed-route-limit 5 set protocols bgp group internal-peers type internal set protocols bgp group internal-peers local-address 192.168.0.1 set protocols bgp group internal-peers export local-direct set protocols bgp group internal-peers neighbor 192.168.0.3 set protocols bgp group external-peers type external set protocols bgp group external-peers export local-direct set protocols bgp group external-peers peer-as 64511 set protocols bgp group external-peers neighbor 10.10.10.2 set protocols ospf area 0.0.0.0 interface fe-1/2/1.0 set protocols ospf area 0.0.0.0 interface fe-1/2/0.0 set protocols ospf area 0.0.0.0 interface lo0.0 passive set policy-options policy-statement local-direct from protocol [local direct] set policy-options policy-statement local-direct then accept </pre>

```
set routing-options autonomous-system 64510
set routing-options router-id 192.168.0.1
```

**Device R2**

```
set interfaces fe-1/2/1 unit 0 description to-R1
set interfaces fe-1/2/1 unit 0 family inet address 10.10.10.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers export local-direct
set protocols bgp group external-peers peer-as 64510
set protocols bgp group external-peers neighbor 10.10.10.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set policy-options policy-statement local-direct from protocol [local direct]
set policy-options policy-statement local-direct then accept
set routing-options autonomous-system 64511
set routing-options router-id 192.168.10.2
```

**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 BGP error handling:

1. Configure the router interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/1 unit 0 description to-R2
user@R1# set fe-1/2/1 unit 0 family inet address 10.10.10.1/30

user@R1# set fe-1/2/0 unit 0 description to-R0
user@R1# set fe-1/2/0 unit 0 family inet address 172.16.10.6/30

user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure an interior gateway protocol (IGP), such as OSPF or IS-IS.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/1.0
user@R1# set interface fe-1/2/0.0
user@R1# set interface lo0.0 passive
```

3. Configure the autonomous system (AS) number and router ID.

```
[edit routing-options]
user@R1# set autonomous-system 64510
user@R1# set router-id 192.168.0.1
```

4. Configure the routing policy.

```
[edit policy-options policy-statement local-direct]
user@R1# set from protocol [local direct]
user@R1# set then accept
```

5. Configure the EBGP session.

```
[edit protocols bgp group external-peers]
user@R1# set type external
user@R1# set export local-direct
```

```
user@R1# set peer-as 64511
user@R1# set neighbor 10.10.10.2
```

6. Configure the IBGP sessions.

```
[edit protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.0.1
user@R1# set export local-direct
user@R1# set neighbor 192.168.0.3
```

7. Enable BGP error tolerance.

```
[edit protocols bgp]
user@R1# set bgp-error-tolerance
```

8. (Optional) Configure the log interval.

```
[edit protocols bgp bgp-error-tolerance]
user@R1# set malformed-update-log-interval 10
```

9. (Optional) Configure a limit for the number of hidden routes to store.

```
[edit protocols bgp bgp-error-tolerance]
user@R1# set malformed-route-limit 5
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 0 {
    description to-R0;
    family inet {
      address 172.16.10.6/30;
    }
  }
}
fe-1/2/1 {
  unit 0 {
    description to-R2;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}

user@R1# show protocols
bgp {
```

```
bgp-error-tolerance {
    malformed-update-log-interval 10;
    malformed-route-limit 5;
}
group internal-peers {
    type internal;
    local-address 192.168.0.1;
    export local-direct;
    neighbor 192.168.0.3;
}
group external-peers {
    type external;
    export local-direct;
    peer-as 64511;
    neighbor 10.10.10.2;
}
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.0;
        interface fe-1/2/0.0;
        interface lo0.0 {
            passive;
        }
    }
}
}

user@R1# show policy-options
policy-statement local-direct {
    from protocol [local direct];
    then accept;
}

user@R1# show routing-options
router-id 192.168.0.1;
autonomous-system 64510;
```

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

---

### Verification

Confirm that the configuration is working properly.

- [Checking the BGP Neighbor Sessions on page 532](#)
- [Checking Hidden Routes on page 534](#)
- [Verifying the Source of the Hidden Routes on page 535](#)

#### ***Checking the BGP Neighbor Sessions***

**Purpose** Verify that BGP error tolerance is enabled, and display the counters related to malformed path attributes.

```

Action user@R1# show bgp neighbor
Peer: 10.10.10.2+50058 AS 64511 Local: 10.10.10.1+179 AS 64510
  Type: External      State: Established      Flags: <Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Export: [ local-direct ]
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Malformed attributes      log interval: 10      route limit: 5
    Attribute:              ORIGIN(1) Last Received: 0 Total Received: 3
    Attribute:              LOCAL_PREF(5) Last Received: 0 Total Received: 2
  Peer ID: 192.168.10.2      Local ID: 192.168.10.1      Active Holdtime: 90
  Keepalive Interval: 30      Group index: 0      Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/1.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 64511)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          0
    Received prefixes:        3
    Accepted prefixes:        0
    Suppressed due to damping: 0
    Advertised prefixes:      2
  Last traffic (seconds): Received 25      Sent 17      Checked 73
  Input messages: Total 2702      Updates 10      Refreshes 0      Octets 51652
  Output messages: Total 2701      Updates 6      Refreshes 0      Octets 51571
  Output Queue[0]: 0

Peer: 192.168.10.3+179 AS 64510 Local: 192.168.10.1+51127 AS 64510
  Type: Internal      State: Established      Flags: <Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Export: [ local-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.10.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Malformed attributes      log interval: 10      route limit: 5
  Peer ID: 192.168.10.3      Local ID: 192.168.10.1      Active Holdtime: 90
  Keepalive Interval: 30      Group index: 1      Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast

```

```
NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 64510)
Peer does not support Addpath
Table inet.0 Bit: 10001
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        3
  Accepted prefixes:        0
  Suppressed due to damping: 0
  Advertised prefixes:      2
Last traffic (seconds): Received 5   Sent 24   Checked 51
Input messages:  Total 417   Updates 3     Refreshes 0     Octets 8006
Output messages: Total 421   Updates 2     Refreshes 0     Octets 8136
Output Queue[0]: 0
```

**Meaning** The Malformed attributes field shows that error tolerance is enabled. The log interval and route limit fields display the configured values.

The attribute counters show that on the EBGP connection, several malformed attributes were received from Device R2.

#### *Checking Hidden Routes*

**Purpose** View information about hidden routes and learn why they are hidden.

```

Action  user@R1> show route hidden detail
inet.0: 42 destinations, 45 routes (36 active, 0 holddown, 6 hidden)
10.0.0.0/32 (1 entry, 0 announced)
    BGP
        Next hop type: Router
        Address: 0x93d8b0c
        Next-hop reference count: 5
        Source: 10.10.10.2
        Next hop type: Router, Next hop index: 782
        Next hop: via fe-1/2/1.0, selected
        Session Id: 0x1
        State: <Hidden Ext>
        Local AS: 1 Peer AS: 1
        Age: 5:32 Metric2: 1
        Validation State: unverified
        Task: BGP_1.10.10.5.62+56218
        AS path: I (MalformedAttr)
        Router ID: 192.168.0.2

10.0.0.1/32 (1 entry, 0 announced)
    BGP
        Next hop type: Router
        Address: 0x93d8b0c
        Next-hop reference count: 5
        Source: 10.10.10.2
        Next hop type: Router, Next hop index: 782
        Next hop: via fe-1/2/1.0, selected
        Session Id: 0x1
        Indirect next hop: 953c000 - INH Session ID: 0x3
        State: <Hidden Int Ext>
        Local AS: 1 Peer AS: 1
        Age: 5:32 Metric2: 1
        Validation State: unverified
        Task: BGP_1.10.10.5.62+56218
        AS path: I (MalformedAttr)
        Router ID: 192.168.0.2

```

**Meaning** The malformed hidden routes are marked with MalformedAttr in the AS path field.

You can remove the hidden routes by running the `clear bgp neighbor 10.10.10.2 malformed-route` command.

#### *Verifying the Source of the Hidden Routes*

**Purpose** View information about hidden routes and learn why they are hidden.

**Action**    user@R1> [show route receive-protocol bgp 10.10.10.2 detail hidden](#)  
inet.0: 42 destinations, 45 routes (36 active, 0 holddown, 6 hidden)  
  10.0.0.0/32 (1 entry, 0 announced)  
    Nexthop: 10.10.10.2  
    Localpref: 100  
    AS path: I (**MalformedAttr**)  
  
  10.0.0.1/32 (1 entry, 0 announced)  
    Nexthop: 10.10.10.2  
    Localpref: 100  
    AS path: I (**MalformedAttr**)

**Meaning**    Junos OS displays MalformedAttr in the AS path field in the output of the [show route receive-protocol bgp 10.10.10.2 detail hidden](#) command.

You can remove the hidden routes by running the [clear bgp neighbor 10.10.10.2 malformed-route](#) command.

**Related Documentation**

- [Example: Preventing BGP Session Resets on page 497](#)
- [Examples: Configuring BGP Flap Damping on page 505](#)

# Multiprotocol BGP Configuration

- [Examples: Configuring Multiprotocol BGP on page 537](#)
- [Example: Configuring Flow Routes on page 550](#)

## Examples: Configuring Multiprotocol BGP

---

- [Understanding Multiprotocol BGP on page 537](#)
- [Example: Configuring IPv6 BGP Routes over IPv4 Transport on page 543](#)
- [Enabling Layer 2 VPN and VPLS Signaling on page 549](#)

## Understanding Multiprotocol BGP

Multiprotocol BGP (MP-BGP) is an extension to BGP that enables BGP to carry routing information for multiple network layers and address families. MP-BGP can carry the unicast routes used for multicast routing separately from the routes used for unicast IP forwarding.

To enable MP-BGP, you configure BGP to carry network layer reachability information (NLRI) for address families other than unicast IPv4 by including the **family inet** statement:

```
family inet {
  (any | flow | labeled-unicast | multicast | unicast) {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    rib-group group-name;
    topology name {
      community {
        target identifier;
      }
    }
  }
}
```

To enable MP-BGP to carry NLRI for the IPv6 address family, include the **family inet6** statement:

```
family inet6 {
  (any | labeled-unicast | multicast | unicast) {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    rib-group group-name;
  }
}
```

On routers only, to enable MP-BGP to carry Layer 3 virtual private network (VPN) NLRI for the IPv4 address family, include the **family inet-vpn** statement:

```
family inet-vpn {
  (any | flow | multicast | unicast) {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    rib-group group-name;
  }
}
```

On routers only, to enable MP-BGP to carry Layer 3 VPN NLRI for the IPv6 address family, include the **family inet6-vpn** statement:

```
family inet6-vpn {
  (any | multicast | unicast) {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    rib-group group-name;
  }
}
```

On routers only, to enable MP-BGP to carry multicast VPN NLRI for the IPv4 address family and to enable VPN signaling, include the **family inet-mvpn** statement:

```

family inet-mvpn {
  signaling {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
  }
}

```

To enable MP-BGP to carry multicast VPN NLRI for the IPv6 address family and to enable VPN signaling, include the **family inet6-mvpn** statement:

```

family inet6-mvpn {
  signaling {
    accepted-prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    <loops number>;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout <forever | minutes>;
    }
  }
}

```

For more information about multiprotocol BGP-based multicast VPNs, see the *Multicast Protocols Feature Guide for Routing Devices*.

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.



**NOTE:** If you change the address family specified in the [edit protocols bgp family] hierarchy level, all current BGP sessions on the routing device are dropped and then reestablished.

In Junos OS Release 9.6 and later, you can specify a loops value for a specific BGP address family.

By default, BGP peers carry only unicast routes used for unicast forwarding purposes. To configure BGP peers to carry only multicast routes, specify the **multicast** option. To configure BGP peers to carry both unicast and multicast routes, specify the **any** option.

When MP-BGP is configured, BGP installs the MP-BGP routes into different routing tables. Each routing table is identified by the protocol family or address family indicator (AFI) and a subsequent address family identifier (SAFI).

The following list shows all possible AFI and SAFI combinations:

- AFI=1, SAFI=1, IPv4 unicast
- AFI=1, SAFI=2, IPv4 multicast
- AFI=1, SAFI=128, L3VPN IPv4 unicast
- AFI=1, SAFI=129, L3VPN IPv4 multicast
- AFI=2, SAFI=1, IPv6 unicast
- AFI=2, SAFI=2, IPv6 multicast
- AFI=25, SAFI=65, BGP-VPLS/BGP-L2VPN
- AFI=2, SAFI=128, L3VPN IPv6 unicast
- AFI=2, SAFI=129, L3VPN IPv6 multicast
- AFI=1, SAFI=132, RT-Constrain
- AFI=1, SAFI=133, Flow-spec
- AFI=1, SAFI=134, Flow-spec
- AFI=3, SAFI=128, CLNS VPN
- AFI=1, SAFI=5, NG-MVPN IPv4
- AFI=2, SAFI=5, NG-MVPN IPv6
- AFI=1, SAFI=66, MDT-SAFI
- AFI=1, SAFI=4, labeled IPv4
- AFI=2, SAFI=4, labeled IPv6 (6PE)

Routes installed in the inet.2 routing table can only be exported to MP-BGP peers because they use the SAFI, identifying them as routes to multicast sources. Routes installed in the inet.0 routing table can only be exported to standard BGP peers.

The inet.2 routing table should be a subset of the routes that you have in inet.0, since it is unlikely that you would have a route to a multicast source to which you could not send unicast traffic. The inet.2 routing table stores the unicast routes that are used for multicast reverse-path-forwarding checks and the additional reachability information learned by MP-BGP from the NLRI multicast updates. An inet.2 routing table is automatically created when you configure MP-BGP (by setting NLRI to **any**).

When you enable MP-BGP, you can do the following:

- [Limiting the Number of Prefixes Received on a BGP Peer Session on page 541](#)
- [Limiting the Number of Prefixes Accepted on a BGP Peer Session on page 541](#)
- [Configuring BGP Routing Table Groups on page 542](#)
- [Resolving Routes to PE Routing Devices Located in Other ASs on page 542](#)
- [Allowing Labeled and Unlabeled Routes on page 543](#)

### Limiting the Number of Prefixes Received on a BGP Peer Session

You can limit the number of prefixes received on a BGP peer session, and log rate-limited messages when the number of injected prefixes exceeds a set limit. You can also tear down the peering when the number of prefixes exceeds the limit.

To configure a limit to the number of prefixes that can be received on a BGP session, include the **prefix-limit** statement:

```
prefix-limit {
  maximum number;
  teardown <percentage> <idle-timeout (forever | minutes)>;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For **maximum *number***, specify a value in the range from 1 through 4,294,967,295. When the specified maximum number of prefixes is exceeded, a system log message is sent.

If you include the **teardown** statement, the session is torn down when the maximum number of prefixes is exceeded. If you specify a percentage, messages are logged when the number of prefixes exceeds that percentage of the specified maximum limit. After the session is torn down, it is reestablished in a short time (unless you include the **idle-timeout** statement). If you include the **idle-timeout** statement, the session can be kept down for a specified amount of time, or forever. If you specify **forever**, the session is reestablished only after the you issue a **clear bgp neighbor** command.



**NOTE:** In Junos OS Release 9.2 and later, you can alternatively configure a limit to the number of prefixes that can be accepted on a BGP peer session. For more information, see “[Understanding Multiprotocol BGP](#)” on page 537.

### Limiting the Number of Prefixes Accepted on a BGP Peer Session

In Junos OS Release 9.2 and later, you can limit the number of prefixes that can be accepted on a BGP peer session. When that specified limit is exceeded, a system log message is sent. You can also specify to reset the BGP session if the limit to the number of specified prefixes is exceeded.

To configure a limit to the number of prefixes that can be accepted on a BGP peer session, include the **accepted-prefix-limit** statement:

```
accepted-prefix-limit {
  maximum number;
  teardown <percentage> <idle-timeout (forever | minutes)>;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For **maximum *number***, specify a value in the range from 1 through 4,294,967,295.

Include the **teardown** statement to reset the BGP peer session when the number of accepted prefixes exceeds the configured limit. You can also include a percentage value from 1 through 100 to have a system log message sent when the number of accepted prefixes exceeds that percentage of the maximum limit. By default, a BGP session that is reset is reestablished within a short time. Include the **idle-timeout** statement to prevent the BGP session from being reestablished for a specified period of time. You can configure a timeout value from 1 through 2400 minutes. Include the **forever** option to prevent the BGP session from being reestablished until you issue the **clear bgp neighbor** command.



**NOTE:** When nonstop active routing (NSR) is enabled and a switchover to a backup Routing Engine occurs, BGP peers that are down are automatically restarted. The peers are restarted even if the **idle-timeout forever** statement is configured.



**NOTE:** Alternatively, you can configure a limit to the number of prefixes that can be *received* (as opposed to accepted) on a BGP peer session. For more information, see [“Limiting the Number of Prefixes Received on a BGP Peer Session”](#) on page 541.

---

### Configuring BGP Routing Table Groups

When a BGP session receives a unicast or multicast NLRI, it installs the route in the appropriate table (**inet.0** or **inet6.0** for unicast, and **inet.2** or **inet6.2** for multicast). To add unicast prefixes to both the unicast and multicast tables, you can configure BGP routing table groups. This is useful if you cannot perform multicast NLRI negotiation.

To configure BGP routing table groups, include the **rib-group** statement:

```
rib-group group-name;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

---

### Resolving Routes to PE Routing Devices Located in Other ASs

You can allow labeled routes to be placed in the **inet.3** routing table for route resolution. These routes are then resolved for provider edge (PE) routing device connections where the remote PE is located across another autonomous system (AS). For a PE routing device to install a route in the VPN routing and forwarding (VRF) routing instance, the next hop must resolve to a route stored within the **inet.3** table.

To resolve routes into the **inet.3** routing table, include the **resolve-vpn** statement:

```
resolve-vpn group-name;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

### Allowing Labeled and Unlabeled Routes

You can allow both labeled and unlabeled routes to be exchanged in a single session. The labeled routes are placed in the inet.3 or inet6.3 routing table, and both labeled and unlabeled unicast routes can be sent to or received by the routing device.

To allow both labeled and unlabeled routes to be exchanged, include the **rib** statement:

```
rib (inet.3 | inet6.3);
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

### Example: Configuring IPv6 BGP Routes over IPv4 Transport

This example demonstrates how to export both IPv6 and IPv4 prefixes over an IPv4 connection where both sides are configured with an IPv4 interface.

- [Requirements on page 543](#)
- [Overview on page 543](#)
- [Configuration on page 544](#)
- [Verification on page 547](#)

#### Requirements

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

#### Overview

Keep the following in mind when exporting IPv6 BGP prefixes:

- BGP derives next-hop prefixes using the IPv4-compatible IPv6 prefix. For example, the IPv4 next-hop prefix **10.19.1.1** translates to the IPv6 next-hop prefix **::ffff:10.19.1.1**.

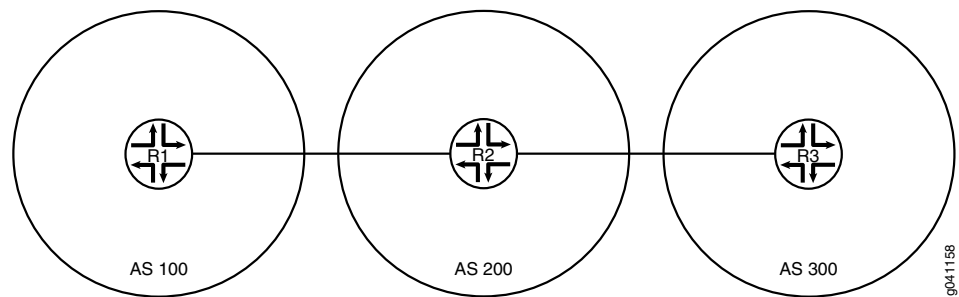


**NOTE:** There must be an active route to the IPv4-compatible IPv6 next hop to export IPv6 BGP prefixes.

- An IPv6 connection must be configured over the link. The connection must be either an IPv6 tunnel or a dual-stack configuration. Dual stacking is used in this example.
- When configuring IPv4-compatible IPv6 prefixes, use a mask that is longer than 96 bits.
- Configure a static route if you want to use normal IPv6 prefixes. This example uses static routes.

[Figure 67 on page 544](#) shows the sample topology.

Figure 67: Topology for Configuring IPv6 BGP Routes over IPv4 Transport



### 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 192.168.10.1/24
set interfaces fe-1/2/0 unit 1 family inet6 address ::ffff:192.168.10.1/120
set interfaces lo0 unit 1 family inet address 10.10.10.1/32
set protocols bgp group ext type external
set protocols bgp group ext family inet unicast
set protocols bgp group ext family inet6 unicast
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 192.168.10.10
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options rib inet6.0 static route ::ffff:192.168.20.0/120 next-hop
::ffff:192.168.10.10
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.10
set routing-options autonomous-system 100

```

**Device R2**

```

set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.10/24
set interfaces fe-1/2/0 unit 2 family inet6 address ::ffff:192.168.10.10/120
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.21/24
set interfaces fe-1/2/1 unit 3 family inet6 address ::ffff:192.168.20.21/120
set interfaces lo0 unit 2 family inet address 10.10.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext family inet unicast
set protocols bgp group ext family inet6 unicast
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext neighbor 192.168.10.1 peer-as 100
set protocols bgp group ext neighbor 192.168.20.1 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options autonomous-system 200

```

```

Device R3    set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
              set interfaces fe-1/2/0 unit 4 family inet6 address ::ffff:192.168.20.1/120
              set interfaces lo0 unit 3 family inet address 10.10.20.1/32
              set protocols bgp group ext type external
              set protocols bgp group ext family inet unicast
              set protocols bgp group ext family inet6 unicast
              set protocols bgp group ext export send-direct
              set protocols bgp group ext export send-static
              set protocols bgp group ext peer-as 200
              set protocols bgp group ext neighbor 192.168.20.21
              set policy-options policy-statement send-direct term 1 from protocol direct
              set policy-options policy-statement send-direct term 1 then accept
              set policy-options policy-statement send-static term 1 from protocol static
              set policy-options policy-statement send-static term 1 then accept
              set routing-options rib inet6.0 static route ::ffff:192.168.10.0/120 next-hop
                ::ffff:192.168.20.21
              set routing-options static route 192.168.10.0/24 next-hop 192.168.20.21
              set routing-options autonomous-system 300

```

### Configuring Device R1

**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 Device R1:

1. Configure the interfaces, including both an IPv4 address and an IPv6 address.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 192.168.10.1/24
user@R1# set fe-1/2/0 unit 1 family inet6 address ::ffff:192.168.10.1/120
user@R1# set lo0 unit 1 family inet address 10.10.10.1/32

```

2. Configure EBGp.

```

[edit protocols bgp group ext]
user@R1# set type external
user@R1# set export send-direct
user@R1# set export send-static
user@R1# set peer-as 200
user@R1# set neighbor 192.168.10.10

```

3. Enable BGP to carry IPv4 unicast and IPv6 unicast routes. .

```

[edit protocols bgp group ext]
user@R1# set family inet unicast
user@R1# set family inet6 unicast

```

IPv4 unicast routes are enabled by default. The configuration is shown here for completeness.

4. Configure the routing policy.

```

[edit policy-options]
user@R1# set policy-statement send-direct term 1 from protocol direct
user@R1# set policy-statement send-direct term 1 then accept
user@R1# set policy-statement send-static term 1 from protocol static

```

```
user@R1# set policy-statement send-static term 1 then accept
```

5. Configure some static routes.

```
[edit routing-options]
user@R1# set rib inet6.0 static route ::ffff:192.168.20.0/120 next-hop
::ffff:192.168.10.10
user@R1# set static route 192.168.20.0/24 next-hop 192.168.10.10
```

6. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 100
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 192.168.10.1/24;
    }
    family inet6 {
      address ::ffff:192.168.10.1/120;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.10.10.1/32;
    }
  }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}
```

```
user@R1# show protocols
bgp {
  group ext {
    type external;
    family inet {
```

```

        unicast;
    }
    family inet6 {
        unicast;
    }
    export [ send-direct send-static ];
    peer-as 200;
    neighbor 192.168.10.10;
}
}

user@R1# show routing-options
rib inet6.0 {
    static {
        route ::ffff:192.168.20.0/120 next-hop ::ffff:192.168.10.10;
    }
}
static {
    route 192.168.20.0/24 next-hop 192.168.10.10;
}
autonomous-system 100;

```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on Device R2 and Device R3, changing the interface names and IP addresses, as needed.

### Verification

Confirm that the configuration is working properly.

- [Checking the Neighbor Status on page 547](#)
- [Checking the Routing Table on page 549](#)

#### *Checking the Neighbor Status*

**Purpose** Make sure that BGP is enabled to carry IPv6 unicast routes.

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

```

user@R2> show bgp neighbor
Peer: 192.168.10.1+179 AS 100 Local: 192.168.10.10+54226 AS 200
  Type: External   State: Established   Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct send-static ]
  Options: <Preference AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast inet6-unicast
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.1      Local ID: 10.10.0.1      Active Holdtime: 90
  Keepalive Interval: 30      Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/0.2
  NLRI for restart configured on peer: inet-unicast inet6-unicast
  NLRI advertised by peer: inet-unicast inet6-unicast
  NLRI for this session: inet-unicast inet6-unicast
  Peer supports Refresh capability (2)

```

```

Stale routes from peer are kept for: 300
Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet-unicast inet6-unicast
NLRI of received end-of-rib markers: inet-unicast inet6-unicast
NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
Peer supports 4 byte AS extension (peer-as 100)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        3
  Accepted prefixes:        2
  Suppressed due to damping: 0
  Advertised prefixes:      4
Table inet6.0 Bit: 20000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        1
  Accepted prefixes:        1
  Suppressed due to damping: 0
  Advertised prefixes:      2
Last traffic (seconds): Received 24   Sent 12   Checked 60
Input messages: Total 132   Updates 6       Refreshes 0       Octets 2700
Output messages: Total 133   Updates 3       Refreshes 0       Octets 2772
Output Queue[0]: 0
Output Queue[1]: 0

Peer: 192.168.20.1+179 AS 300 Local: 192.168.20.21+54706 AS 200
Type: External State: Established Flags: <Sync>
Last State: OpenConfirm Last Event: RecvKeepAlive
Last Error: None
Export: [ send-direct send-static ]
Options: <Preference AddressFamily PeerAS Refresh>
Address families configured: inet-unicast inet6-unicast
Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.10.20.1 Local ID: 10.10.0.1 Active Holdtime: 90
Keepalive Interval: 30 Peer index: 1
BFD: disabled, down
Local Interface: fe-1/2/1.3
NLRI for restart configured on peer: inet-unicast inet6-unicast
NLRI advertised by peer: inet-unicast inet6-unicast
NLRI for this session: inet-unicast inet6-unicast
Peer supports Refresh capability (2)
Stale routes from peer are kept for: 300
Peer does not support Restarter functionality
NLRI that restart is negotiated for: inet-unicast inet6-unicast
NLRI of received end-of-rib markers: inet-unicast inet6-unicast
NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
Peer supports 4 byte AS extension (peer-as 300)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        3
  Accepted prefixes:        2
  Suppressed due to damping: 0
  Advertised prefixes:      4

```

```

Table inet6.0 Bit: 20000
RIB State: BGP restart is complete
Send state: in sync
Active prefixes:          0
Received prefixes:        1
Accepted prefixes:        1
Suppressed due to damping: 0
Advertised prefixes:      2
Last traffic (seconds): Received 1   Sent 15   Checked 75
Input messages:  Total 133   Updates 6   Refreshes 0   Octets 2719
Output messages: Total 131   Updates 3   Refreshes 0   Octets 2734
Output Queue[0]: 0
Output Queue[1]: 0

```

**Meaning** The various occurrences of **inet6-unicast** in the output shows that BGP is enabled to carry IPv6 unicast routes.

### Checking the Routing Table

**Purpose** Make sure that Device R2 has BGP routes in its inet6.0 routing table.

**Action** From operational mode, enter the **show route protocol bgp inet6.0** command.

```

user@R2> show route protocol bgp table inet6.0
inet6.0: 7 destinations, 10 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

::ffff:192.168.10.0/120 [BGP/170] 01:03:49, localpref 100, from 192.168.20.1
                        AS path: 300 I
                        > to ::ffff:192.168.20.21 via fe-1/2/1.3
::ffff:192.168.20.0/120 [BGP/170] 01:03:53, localpref 100, from 192.168.10.1
                        AS path: 100 I
                        > to ::ffff:192.168.10.10 via fe-1/2/0.2

```

## Enabling Layer 2 VPN and VPLS Signaling

You can enable BGP to carry Layer 2 VPN and VPLS NLRI messages.

To enable VPN and VPLS signaling, include the **family** statement:

```

family {
  l2vpn {
    signaling {
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
    }
  }
}

```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

To configure a maximum number of prefixes, include the **prefix-limit** statement:

```

prefix-limit {

```

```
    maximum number;  
    teardown <percentage> <idle-timeout (forever | minutes)>;  
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

When you set the maximum number of prefixes, a message is logged when that number is reached. If you include the **teardown** statement, the session is torn down when the maximum number of prefixes is reached. If you specify a percentage, messages are logged when the number of prefixes reaches that percentage. Once the session is torn down, it is reestablished in a short time. Include the **idle-timeout** statement to keep the session down for a specified amount of time, or forever. If you specify **forever**, the session is reestablished only after you use the **clear bgp neighbor** command.

#### Related Documentation

- [Example: Configuring Flow Routes on page 550](#)
- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

---

## Example: Configuring Flow Routes

- [Understanding Flow Routes on page 550](#)
- [Example: Enabling BGP to Carry Flow-Specification Routes on page 554](#)

### Understanding Flow Routes

A flow route is an aggregation of match conditions for IP packets. Flow routes are propagated through the network using flow-specification network-layer reachability information (NLRI) messages and installed into the flow routing table **instance-name.inetflow.0**. Packets can travel through flow routes only if specific match conditions are met.

Flow routes and firewall filters are similar in that they filter packets based on their components and perform an action on the packets that match. Flow routes provide traffic filtering and rate-limiting capabilities much like firewall filters. In addition, you can propagate flow routes across different autonomous systems.

Flow routes are propagated by BGP through flow-specification NLRI messages. You must enable BGP to propagate these NLRIs.

---

#### Match Conditions for Flow Routes

You specify conditions that the packet must match before the action in the **then** statement is taken for a flow route. All conditions in the **from** statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

To configure a match condition, include the **match** statement at the **[edit routing-options flow]** hierarchy level.

[Table 7 on page 551](#) describes the flow route match conditions.

Table 7: Flow Route Match Conditions

Match Condition	Description
<b>destination prefix</b>	IP destination address field.
<b>destination-port number</b>	<p>TCP or User Datagram Protocol (UDP) destination port field. You cannot specify both the <b>port</b> and <b>destination-port</b> match conditions in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): <b>afs</b> (1483), <b>bgp</b> (179), <b>biff</b> (512), <b>bootpc</b> (68), <b>bootps</b> (67), <b>cmd</b> (514), <b>cvspserver</b> (2401), <b>dhcp</b> (67), <b>domain</b> (53), <b>eklogin</b> (2105), <b>ekshell</b> (2106), <b>exec</b> (512), <b>finger</b> (79), <b>ftp</b> (21), <b>ftp-data</b> (20), <b>http</b> (80), <b>https</b> (443), <b>ident</b> (113), <b>imap</b> (143), <b>kerberos-sec</b> (88), <b>klogin</b> (543), <b>kpasswd</b> (761), <b>krb-prop</b> (754), <b>krbupdate</b> (760), <b>kshell</b> (544), <b>ldap</b> (389), <b>login</b> (513), <b>mobileip-agent</b> (434), <b>mobilip-mn</b> (435), <b>msdp</b> (639), <b>netbios-dgm</b> (138), <b>netbios-ns</b> (137), <b>netbios-ssn</b> (139), <b>nfsd</b> (2049), <b>nnpt</b> (119), <b>ntalk</b> (518), <b>ntp</b> (123), <b>pop3</b> (110), <b>pptp</b> (1723), <b>printer</b> (515), <b>radacct</b> (1813), <b>radius</b> (1812), <b>rip</b> (520), <b>rkinit</b> (2108), <b>smtp</b> (25), <b>snmp</b> (161), <b>snmptrap</b> (162), <b>snpp</b> (444), <b>socks</b> (1080), <b>ssh</b> (22), <b>sunrpc</b> (111), <b>syslog</b> (514), <b>tacacs-ds</b> (65), <b>talk</b> (517), <b>telnet</b> (23), <b>tftp</b> (69), <b>timed</b> (525), <b>who</b> (513), <b>xdmcp</b> (177), <b>zephyr-clt</b> (2103), or <b>zephyr-hm</b> (2104).</p>
<b>dscp number</b>	<p>Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant six bits of this byte form the DSCP.</p> <p>You can specify DSCP in hexadecimal or decimal form.</p>
<b>fragment type</b>	<p>Fragment type field. The keywords are grouped by the fragment type with which they are associated:</p> <ul style="list-style-type: none"> <li>• <b>dont-fragment</b></li> <li>• <b>first-fragment</b></li> <li>• <b>is-fragment</b></li> <li>• <b>last-fragment</b></li> <li>• <b>not-a-fragment</b></li> </ul>
<b>icmp-code number</b>	<p>ICMP code field. This value or keyword provides more specific information than <b>icmp-type</b>. Because the value's meaning depends upon the associated <b>icmp-type</b> value, you must specify <b>icmp-type</b> along with <b>icmp-code</b>.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> <li>• parameter-problem: <b>ip-header-bad</b> (0), <b>required-option-missing</b> (1)</li> <li>• redirect: <b>redirect-for-host</b> (1), <b>redirect-for-network</b> (0), <b>redirect-for-tos-and-host</b> (3), <b>redirect-for-tos-and-net</b> (2)</li> <li>• time-exceeded: <b>tll-eq-zero-during-reassembly</b> (1), <b>tll-eq-zero-during-transit</b> (0)</li> <li>• unreachable: <b>communication-prohibited-by-filtering</b> (13), <b>destination-host-prohibited</b> (10), <b>destination-host-unknown</b> (7), <b>destination-network-prohibited</b> (9), <b>destination-network-unknown</b> (6), <b>fragmentation-needed</b> (4), <b>host-precedence-violation</b> (14), <b>host-unreachable</b> (1), <b>host-unreachable-for-TOS</b> (12), <b>network-unreachable</b> (0), <b>network-unreachable-for-TOS</b> (11), <b>port-unreachable</b> (3), <b>precedence-cutoff-in-effect</b> (15), <b>protocol-unreachable</b> (2), <b>source-host-isolated</b> (8), <b>source-route-failed</b> (5)</li> </ul>

Table 7: Flow Route Match Conditions (*continued*)

Match Condition	Description
<b>icmp-type number</b>	<p>ICMP packet type field. Normally, you specify this match in conjunction with the <b>protocol</b> match statement to determine which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): <b>echo-reply</b> (0), <b>echo-request</b> (8), <b>info-reply</b> (16), <b>info-request</b> (15), <b>mask-request</b> (17), <b>mask-reply</b> (18), <b>parameter-problem</b> (12), <b>redirect</b> (5), <b>router-advertisement</b> (9), <b>router-solicit</b> (10), <b>source-quench</b> (4), <b>time-exceeded</b> (11), <b>timestamp</b> (13), <b>timestamp-reply</b> (14), or <b>unreachable</b> (3).</p>
<b>packet-length number</b>	Total IP packet length.
<b>port number</b>	<p>TCP or UDP source or destination port field. You cannot specify both the <b>port</b> match and either the <b>destination-port</b> or <b>source-port</b> match condition in the same term.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed under <b>destination-port</b>.</p>
<b>protocol number</b>	<p>IP protocol field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): <b>ah</b>, <b>egp</b> (8), <b>esp</b> (50), <b>gre</b> (47), <b>icmp</b> (1), <b>igmp</b> (2), <b>ipip</b> (4), <b>ipv6</b> (41), <b>ospf</b> (89), <b>pim</b> (103), <b>rsvp</b> (46), <b>tcp</b> (6), or <b>udp</b> (17).</p>
<b>source prefix</b>	IP source address field.
<b>source-port number</b>	<p>TCP or UDP source port field. You cannot specify the <b>port</b> and <b>source-port</b> match conditions in the same term.</p> <p>In place of the numeric field, you can specify one of the text synonyms listed under <b>destination-port</b>.</p>
<b>tcp-flag type</b>	TCP header format.

### Actions for Flow Routes

You can specify the action to take if the packet matches the conditions you have configured in the flow route. To configure an action, include the **then** statement at the **[edit routing-options flow]** hierarchy level.

Table 8 on page 552 describes the flow route actions.

Table 8: Flow Route Action Modifiers

Action or Action Modifier	Description
<b>Actions</b>	
<b>accept</b>	Accept a packet. This is the default.
<b>discard</b>	Discard a packet silently, without sending an Internet Control Message Protocol (ICMP) message.
<b>community</b>	Replace any communities in the route with the specified communities.
<b>next-term</b>	Continue to the next match condition for evaluation.

Table 8: Flow Route Action Modifiers (*continued*)

Action or Action Modifier	Description
<b>routing-instance</b> <i>extended-community</i>	Specify a routing instance to which packets are forwarded.
<b>rate-limit</b> <i>bits-per-second</i>	Limit the bandwidth on the flow route. Express the limit in bits per second (bps).
<b>sample</b>	Sample the traffic on the flow route.

### Validating Flow Routes

The Junos OS installs flow routes into the flow routing table only if they have been validated using the validation procedure. The Routing Engine does the validation before the installing routes into the flow routing table.

Flow routes received using the BGP network layer reachability information (NLRI) messages are validated before they are installed into the flow primary instance routing table **instance.inetflow.0**. The validation procedure is described in the draft-ietf-idr-flow-spec-09.txt, *Dissemination of Flow Specification Rules*. You can bypass the validation process for flow routes using BGP NLRI messages and use your own specific import policy.

To trace validation operations, include the **validation** statement at the **[edit routing-options flow]** hierarchy level.

### Support for BGP Flow-Specification Algorithm Version 7 and Later

By default, the Junos OS uses the term-ordering algorithm defined in version 6 of the BGP flow specification draft. In Junos OS Release 10.0 and later, you can configure the router to comply with the term-ordering algorithm first defined in version 7 of the BGP flow specification and supported through RFC 5575, *Dissemination of Flow Specification Routes*.



**BEST PRACTICE:** We recommend that you configure the Junos OS to use the term-ordering algorithm first defined in version 7 of the BGP flow specification draft. We also recommend that you configure the Junos OS to use the same term-ordering algorithm on all routing instances configured on a router.

To configure BGP to use the flow-specification algorithm first defined in version 7 of the Internet draft, include the **standard** statement at the **[edit routing-options flow term-order]** hierarchy level.

To revert to using the term-ordering algorithm defined in version 6, include the **legacy** statement at the **[edit routing-options flow term-order]** hierarchy level.



**NOTE:** The configured term order has only local significance. That is, the term order does not propagate with flow routes sent to the remote BGP peers, whose term order is completely determined by their own term order configuration. Therefore, you should be careful when configuring the order-dependent action next term when you are not aware of the term order configuration of the remote peers. The local next term might differ from the next term configured on the remote peer.

---

## Example: Enabling BGP to Carry Flow-Specification Routes

This example shows how to allow BGP to carry flow-specification network layer reachability information (NLRI) messages.

- [Requirements on page 554](#)
- [Overview on page 554](#)
- [Configuration on page 556](#)
- [Verification on page 564](#)

### Requirements

---

Before you begin:

- Configure the device interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure a routing policy that exports routes (such as direct routes or IGP routes) from the routing table into BGP.

### Overview

---

Propagating firewall filter information as part of BGP enables you to propagate firewall filters against denial-of-service (DOS) attacks dynamically across autonomous systems. Flow routes are encapsulated into the flow-specification NLRI and propagated through a network or virtual private networks (VPNs), sharing filter-like information. Flow routes are an aggregation of match conditions and resulting actions for packets. They provide you with traffic filtering and rate-limiting capabilities much like firewall filters. Unicast flow routes are supported for the default instance, VPN routing and forwarding (VRF) instances, and virtual-router instances.

Import and export policies can be applied to the family **inet flow** or family **inet-vpn flow** NLRI, affecting the flow routes accepted or advertised, similar to the way import and export policies are applied to other BGP families. The only difference is that the flow policy configuration must include the **from rib inetflow.0** statement. This statement causes the policy to be applied to the flow routes. An exception to this rule occurs if the policy has only the **then reject** or the **then accept** statement and no **from** statement. Then, the policy affects all routes, including IP unicast and IP flow.

The flow route filters are first configured on a router statically, with a set of matching criteria followed by the actions to be taken. Then, in addition to **family inet unicast**, **family inet flow** (or **family inet-vpn flow**) is configured between this BGP-enabled device and its peers.

By default, statically configured flow routes (firewall filters) are advertised to other BGP-enabled devices that support the **family inet flow** or **family inet-vpn flow** NLRI.

The receiving BGP-enabled device performs a validation process before installing the firewall filter into the flow routing table *instance-name.inetflow.0*. The validation procedure is described in RFC 5575, *Dissemination of Flow Specification Rules*.

The receiving BGP-enabled device accepts a flow route if it passes the following criteria:

- The originator of a flow route matches the originator of the best match unicast route for the destination address that is embedded in the route.
- There are no more specific unicast routes, when compared to the destination address of the flow route, for which the active route has been received from a different next-hop autonomous system.

The first criterion ensures that the filter is being advertised by the next-hop used by unicast forwarding for the destination address embedded in the flow route. For example, if a flow route is given as 10.1.1.1, proto=6, port=80, the receiving BGP-enabled device selects the more specific unicast route in the unicast routing table that matches the destination prefix 10.1.1.1/32. On a unicast routing table containing 10.1/16 and 10.1.1/24, the latter is chosen as the unicast route to compare against. Only the active unicast route entry is considered. This follows the concept that a flow route is valid if advertised by the originator of the best unicast route.

The second criterion addresses situations in which a given address block is allocated to different entities. Flows that resolve to a best-match unicast route that is an aggregate route are only accepted if they do not cover more specific routes that are being routed to different next-hop autonomous systems.

You can bypass the validation process and use your own specific import policy. To disable the validation procedure and use an import policy instead, include the **no-validate** statement at the **[edit protocols bgp group group-name family inet flow]** hierarchy level. The import policy configured to select flow routes can only be used to match on a route community. It cannot be configured to match on flow source addresses, destination addresses, ports, or any other information.

After a flow route is installed in the **inetflow.0** table, it is also added to the list of firewall filters in the kernel.

On routers only, flow-specification NLRI messages are supported in VPNs. The VPN compares the route target extended community in the NLRI to the import policy. If there is a match, the VPN can start using the flow routes to filter and rate-limit packet traffic. Received flow routes are installed into the flow routing table *instance-name.inetflow.0*. Flow routes can also be propagated throughout a VPN network and shared among VPNs. To enable multiprotocol BGP (MP-BGP) to carry flow-specification NLRI for the **inet-vpn** address family, include the **flow** statement at the **[edit protocols bgp group group-name**

**family inet-vpn**] hierarchy level. VPN flow routes are supported for the default instance only. Flow routes configured for VPNs with family **inet-vpn** are not automatically validated, so the **no-validate** statement is not supported at the **[edit protocols bgp group group-name family inet-vpn]** hierarchy level. No validation is needed if the flow routes are configured locally between devices in a single AS.

Import and export policies can be applied to the **family inet flow** or **family inet-vpn flow** NLRI, affecting the flow routes accepted or advertised, similar to the way import and export policies are applied to other BGP families. The only difference is that the flow policy configuration must include the **from rib inetflow.0** statement. This statement causes the policy to be applied to the flow routes. An exception to this rule occurs if the policy has only the **then reject** or the **then accept** statement and no **from** statement. Then, the policy affects all routes, including IP unicast and IP flow.

This example shows how to configure the following export policies:

- A policy that allows the advertisement of flow routes specified by a route-filter. Only the flow routes covered by the 10.13/16 block are advertised. This policy does not affect unicast routes.
- A policy that allows all unicast and flow routes to be advertised to the neighbor.
- A policy that disallows all routes (unicast or flow) to be advertised to the neighbor.

---

## Configuration

- [Configuring a Static Flow Route on page 556](#)
- [Advertising Flow Routes Specified by a Route Filter on page 558](#)
- [Advertising All Unicast and Flow Routes on page 559](#)
- [Advertising No Unicast or Flow Routes on page 560](#)
- [Limiting the Number of Flow Routes Installed in a Routing Table on page 562](#)
- [Limiting the Number of Prefixes Received on a BGP Peering Session on page 563](#)

### *Configuring a Static Flow Route*

#### 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.

```
set routing-options flow route block-10.131.1.1 match destination 10.131.1.1/32
set routing-options flow route block-10.131.1.1 match protocol icmp
set routing-options flow route block-10.131.1.1 match icmp-type echo-request
set routing-options flow route block-10.131.1.1 then discard
set routing-options flow term-order standard
```

**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 BGP peer sessions:

1. Configure the match conditions.

```
[edit routing-options flow route block-10.131.1.1]
user@host# set match destination 10.131.1.1/32
user@host# set match protocol icmp
user@host# set match icmp-type echo-request
```

2. Configure the action.

```
[edit routing-options flow route block-10.131.1.1]
user@host# set then discard
```

3. (Recommended) For the flow specification algorithm, configure the standard-based term order.

```
[edit routing-options flow]
user@host# set term-order standard
```

In the default term ordering algorithm, as specified in the flowspec RFC draft Version 6, a term with less specific matching conditions is always evaluated before a term with more specific matching conditions. This causes the term with more specific matching conditions to never be evaluated. Version 7 of RFC 5575 made a revision to the algorithm so that the more specific matching conditions are evaluated before the less specific matching conditions. For backward compatibility, the default behavior is not altered in Junos OS, even though the newer algorithm makes more sense. To use the newer algorithm, include the **term-order standard** statement in the configuration. This statement is supported in Junos OS Release 10.0 and later.

**Results** From configuration mode, confirm your configuration by entering the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show routing-options
flow {
  term-order standard;
  route block-10.131.1.1 {
    match {
      destination 10.131.1.1/32;
      protocol icmp;
      icmp-type echo-request;
    }
    then discard;
  }
}
```

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

**Advertising Flow Routes Specified by a Route Filter**

**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.

```
set protocols bgp group core family inet unicast
set protocols bgp group core family inet flow
set protocols bgp group core export p1
set protocols bgp group core peer-as 65000
set protocols bgp group core neighbor 10.12.99.5
set policy-options policy-statement p1 term a from rib inetflow.0
set policy-options policy-statement p1 term a from route-filter 10.13.0.0/16 orlonger
set policy-options policy-statement p1 term a then accept
set policy-options policy-statement p1 term b then reject
set routing-options autonomous-system 65001
```

**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 BGP peer sessions:

1. Configure the BGP group.

```
[edit protocols bgp group core]
user@host# set family inet unicast
user@host# set family inet flow
user@host# set export p1
user@host# set peer-as 65000
user@host# set neighbor 10.12.99.5
```

2. Configure the flow policy.

```
[edit policy-options policy-statement p1]
user@host# set term a from rib inetflow.0
user@host# set term a from route-filter 10.13.0.0/16 orlonger
user@host# set term a then accept
user@host# set term b then reject
```

3. Configure the local autonomous system (AS) number.

```
[edit routing-options]
user@host# set autonomous-system 65001
```

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

```
[edit]
user@host# show protocols
bgp {
  group core {
    family inet {
```

```

        unicast;
        flow;
    }
    export p1;
    peer-as 65000;
    neighbor 10.12.99.5;
}

[edit]
user@host# show policy-options
policy-statement p1 {
    term a {
        from {
            rib inetflow.0;
            route-filter 10.13.0.0/16 orlonger;
        }
        then accept;
    }
    term b {
        then reject;
    }
}

[edit]
user@host# show routing-options
autonomous-system 65001;

```

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

### ***Advertising All Unicast and Flow Routes***

**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.

```

set protocols bgp group core family inet unicast
set protocols bgp group core family inet flow
set protocols bgp group core export p1
set protocols bgp group core peer-as 65000
set protocols bgp group core neighbor 10.12.99.5
set policy-options policy-statement p1 term a then accept
set routing-options autonomous-system 65001

```

**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 BGP peer sessions:

1. Configure the BGP group.
 

```

[edit protocols bgp group core]
user@host# set family inet unicast
user@host# set family inet flow

```

```
user@host# set export p1
user@host# set peer-as 65000
user@host# set neighbor 10.12.99.5
```

2. Configure the flow policy.

```
[edit policy-options policy-statement p1]
user@host# set term a then accept
```

3. Configure the local autonomous system (AS) number.

```
[edit routing-options]
user@host# set autonomous-system 65001
```

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

```
[edit]
user@host# show protocols
bgp {
  group core {
    family inet {
      unicast;
      flow;
    }
    export p1;
    peer-as 65000;
    neighbor 10.12.99.5;
  }
}

[edit]
user@host# show policy-options
policy-statement p1 {
  term a {
    then accept;
  }
}

[edit]
user@host# show routing-options
autonomous-system 65001;
```

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

### ***Advertising No Unicast or Flow Routes***

**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.

```
set protocols bgp group core family inet unicast
set protocols bgp group core family inet flow
set protocols bgp group core export p1
```

```

set protocols bgp group core peer-as 65000
set protocols bgp group core neighbor 10.12.99.5
set policy-options policy-statement p1 term a then reject
set routing-options autonomous-system 65001

```

**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 BGP peer sessions:

1. Configure the BGP group.
 

```

[edit protocols bgp group core]
user@host# set family inet unicast
user@host# set family inet flow
user@host# set export p1
user@host# set peer-as 65000
user@host# set neighbor 10.12.99.5

```
2. Configure the flow policy.
 

```

[edit policy-options policy-statement p1]
user@host# set term a then reject

```
3. Configure the local autonomous system (AS) number.
 

```

[edit routing-options]
user@host# set autonomous-system 65001

```

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

```

[edit]
user@host# show protocols
bgp {
  group core {
    family inet {
      unicast;
      flow;
    }
    export p1;
    peer-as 65000;
    neighbor 10.12.99.5;
  }
}

[edit]
user@host# show policy-options
policy-statement p1 {
  term a {
    then reject;
  }
}

```

```
[edit]
user@host# show routing-options
autonomous-system 65001;
```

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

### *Limiting the Number of Flow Routes Installed in a Routing Table*

#### 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.

```
set routing-options rib inetflow.0 maximum-prefixes 1000
set routing-options rib inetflow.0 maximum-prefixes threshold 50
```

#### 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*.



**NOTE:** Application of a route limit might result in unpredictable dynamic route protocol behavior. For example, once the limit is reached and routes are being rejected, BGP does not necessarily attempt to reinstall the rejected routes after the number of routes drops below the limit. BGP sessions might need to be cleared to resolve this issue.

To limit the flow routes:

1. Set an upper limit for the number of prefixes installed in **inetflow.0** table.  

```
[edit routing-options rib inetflow.0]
user@host# set maximum-prefixes 1000
```
2. Set a threshold value of 50 percent, where when 500 routes are installed, a warning is logged in the system log.  

```
[edit routing-options rib inetflow.0]
user@host# set maximum-prefixes threshold 50
```

#### Results

From configuration mode, confirm your configuration by entering the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show routing-options
rib inetflow.0 {
  maximum-prefixes 1000 threshold 50;
}
```

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

**Limiting the Number of Prefixes Received on a BGP Peering Session**

**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.

```
set protocols bgp group x1 neighbor 10.12.99.2 family inet flow prefix-limit maximum 1000
set protocols bgp group x1 neighbor 10.12.99.2 family inet flow prefix-limit teardown 50
```

**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*.

Configuring a prefix limit for a specific neighbor provides more predictable control over which peer can advertise how many flow routes.

To limit the number of prefixes:

1. Set a limit of 1000 BGP routes from neighbor 10.12.99.2.

```
[edit protocols bgp group x1]
user@host# set neighbor 10.12.99.2 family inet flow prefix-limit maximum 1000
```

2. Configure the neighbor session to be brought down when the maximum number of prefixes is reached.

```
[edit routing-options rib inetflow.0]
user@host# set neighbor 10.12.99.2 family inet flow prefix-limit teardown 50
```

If you specify a percentage, as shown here, messages are logged when the number of prefixes reaches that percentage.

After the session is brought down, the session reestablishes in a short time unless you include the **idle-timeout** statement.

**Results** From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show protocols
bgp {
  group x1 {
    neighbor 10.12.99.2 {
      flow {
        prefix-limit {
          maximum 1000;
          teardown 50;
        }
      }
    }
  }
}
```

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

### Verification

---

Confirm that the configuration is working properly.

- [Verifying the NLRI on page 564](#)
- [Verifying Routes on page 565](#)
- [Verifying Flow Validation on page 566](#)
- [Verifying Firewall Filters on page 566](#)
- [Verifying System Logging When Exceeding the Number of Allowed Flow Routes on page 567](#)
- [Verifying System Logging When Exceeding the Number of Prefixes Received on a BGP Peering Session on page 567](#)

#### *Verifying the NLRI*

**Purpose** Look at the NLRI enabled for the neighbor.

**Action** From operational mode, run the **show bgp neighbor 10.12.99.5** command. Look for **inet-flow** in the output.

```
user@host> show bgp neighbor 10.12.99.5
Peer: 10.12.99.5+3792 AS 65000 Local: 10.12.99.6+179 AS 65002
Type: External State: Established Flags: <Sync>
Last State: OpenConfirm Last Event: RecvKeepAlive
Last Error: None
Export: [ direct ]
Options: <Preference HoldTime AddressFamily PeerAS Refresh>
Address families configured: inet-unicast inet-multicast inet-flow
Holdtime: 90 Preference: 170
Number of flaps: 1
Error: 'Cease' Sent: 0 Recv: 1
Peer ID: 10.255.71.161 Local ID: 10.255.124.107 Active Holdtime: 90
Keepalive Interval: 30 Peer index: 0
Local Interface: e1-3/0/0.0
NLRI advertised by peer: inet-unicast inet-multicast inet-flow
NLRI for this session: inet-unicast inet-multicast inet-flow
Peer supports Refresh capability (2)
Table inet.0 Bit: 10000
RIB State: BGP restart is complete
Send state: in sync
Active prefixes: 2
Received prefixes: 2
Suppressed due to damping: 0
Advertised prefixes: 3
Table inet.2 Bit: 20000
RIB State: BGP restart is complete
Send state: in sync
Active prefixes: 0
Received prefixes: 0
Suppressed due to damping: 0
Advertised prefixes: 0
Table inetFlow.0 Bit: 30000
RIB State: BGP restart is complete
Send state: in sync
```

```

Active prefixes: 0
Received prefixes: 0
Suppressed due to damping: 0
Advertised prefixes: 0
Last traffic (seconds): Received 29 Sent 15 Checked 15
Input messages: Total 5549 Updates 2618 Refreshes 0 Octets 416486
Output messages: Total 2943 Updates 1 Refreshes 0 Octets 55995
Output Queue[0]: 0
Output Queue[1]: 0
Output Queue[2]: 0

```

### Verifying Routes

**Purpose** Look at the flow routes. The sample output shows a flow route learned from BGP and a statically configured flow route.

For locally configured flow routes (configured at the **[edit routing-options flow]** hierarchy level), the routes are installed by the flow protocol. Therefore, you can display the flow routes by specifying the table, as in **show route table inetflow.0** or **show route table instance-name.inetflow.0**, where *instance-name* is the routing instance name. Or, you can display all locally configured flow routes across multiple routing instances by running the **show route protocol flow** command.

If a flow route is not locally configured, but received from the router's BGP peer, this flow route is installed in the routing table by BGP. You can display the flow routes by specifying the table or by running **show route protocol bgp**, which displays all BGP routes (flow and non-flow).

**Action** From operational mode, run the **show route table inetflow.0** command.

```

user@host> show route table inetflow.0
inetflow.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.12.44.1,*/term:1
                *[Flow/5] 00:04:22
                Fictitious
*,10.12.44.1/term:2
                *[Flow/5] 00:09:34
                Fictitious

```

```

user@host> show route table inetflow.0 extensive
inetflow.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
7.7.7.7,8.8.8.8/term:1 (1 entry, 1 announced)
TSI:
KRT in dfwd;
Action(s): accept,count
    *Flow Preference: 5
           Next hop type: Fictitious
           Address: 0x8d383a4
           Next-hop reference count: 3
           State: <Active>
           Local AS: 65000
           Age: 9:50
           Task: RT Flow
           Announcement bits (1): 0-Flow
           AS path: I

```

**Meaning** A flow route represents a term of a firewall filter. When you configure a flow route, you specify the match conditions and the actions. In the match attributes, you can match a source address, a destination address, and other qualifiers such as the port and the protocol. For a single flow route that contains multiple match conditions, all the match conditions are encapsulated in the prefix field of the route. When you issue the **show route** command on a flow route, the prefix field of the route is displayed with all of the match conditions. **10.12.44.1,\*** means that the matching condition is **match destination 10.12.44.1/32**. If the prefix in the output were **\*10.12.44.1**, this would mean that the match condition was **match source 10.12.44.1/32**. If the matching conditions contain both a source and a destination, the asterisk is replaced with the address.

The term-order numbers indicate the sequence of the terms (flow routes) being evaluated in the firewall filter. The **show route extensive** command displays the actions for each term (route).

#### *Verifying Flow Validation*

**Purpose** Display flow route information.

**Action** From operational mode, run the **show route flow validation detail** command.

```
user@host> show route flow validation detail
inet.0:
0.0.0.0/0
    Internal node: best match, inconsistent
10.0.0.0/8
    Internal node: no match, inconsistent
10.12.42.0/24
    Internal node: no match, consistent, next-as: 65003
    Active unicast route
        Dependent flow destinations: 1
        Origin: 10.255.124.106, Neighbor AS: 65003
10.12.42.1/32
    Flow destination (1 entries, 1 match origin)
        Unicast best match: 10.12.42.0/24
        Flags: Consistent
10.131.0.0/16
    Internal node: no match, consistent, next-as: 65001
    Active unicast route
        Dependent flow destinations: 5000
        Origin: 10.12.99.2, Neighbor AS: 65001
10.131.0.0/19
    Internal node: best match
10.131.0.0/20
    Internal node: best match
10.131.0.0/21
```

#### *Verifying Firewall Filters*

**Purpose** Display the firewall filters that are installed in the kernel.

**Action** From operational mode, run the **show firewall** command.

```
user@host> show firewall
Filter: __default_bpdu_filter__
Filter: __dynamic_default_inet__
Counters:
Name                                     Bytes      Packets
10.12.42.1,*                             0           0
196.1.28/23,*                             0           0
196.1.30/24,*                             0           0
196.1.31/24,*                             0           0
196.1.32/24,*                             0           0
196.1.56/21,*                             0           0
196.1.68/24,*                             0           0
196.1.69/24,*                             0           0
196.1.70/24,*                             0           0
196.1.75/24,*                             0           0
196.1.76/24,*                             0           0
```

#### *Verifying System Logging When Exceeding the Number of Allowed Flow Routes*

**Purpose** If you configure a limit on the number of flow routes installed, as described in [“Limiting the Number of Flow Routes Installed in a Routing Table” on page 562](#), view the system log message when the threshold is reached.

**Action** From operational mode, run the **show log <log-filename>** command.

```
user@host> show log flow-routes-log-file
Jul 12 08:19:01 host rpd[2748]: RPD_RT_MAXROUTES_WARN: Number of routes (1000)
in
table inetflow.0 exceeded warning threshold (50 percent of configured maximum
1000)
```

#### *Verifying System Logging When Exceeding the Number of Prefixes Received on a BGP Peering Session*

**Purpose** If you configure a limit on the number of flow routes installed, as described in [“Limiting the Number of Prefixes Received on a BGP Peering Session” on page 563](#), view the system log message when the threshold is reached.

**Action** From operational mode, run the **show log <log-filename>** command.

```
user@host> show log flow-routes-log-file
Jul 12 08:44:47 host rpd[2748]: 10.12.99.2 (External AS 65001): Shutting down
peer due to
exceeding configured maximum prefix-limit(1000) for inet-flow nlr: 1001
```

**Related Documentation**

- [Examples: Configuring Multiprotocol BGP on page 537](#)



## CHAPTER 13

# BGP CLNS Configuration

- [Example: Configuring BGP and CLNS on page 569](#)

### Example: Configuring BGP and CLNS

---

- [Understanding BGP for CLNS VPNs on page 569](#)
- [Example: Configuring BGP for CLNS VPNs on page 569](#)
- [Enabling BGP to Carry CLNS Routes on page 571](#)

### Understanding BGP for CLNS VPNs

BGP extensions allow BGP to carry Connectionless Network Service (CLNS) virtual private network (VPN) network layer reachability information (NLRI) between provider edge (PE) routers. Each CLNS route is encapsulated into a CLNS VPN NLRI and propagated between remote sites in a VPN.

CLNS is a Layer 3 protocol similar to IP version 4 (IPv4). CLNS uses network service access points (NSAPs) to address end systems. This allows for a seamless autonomous system (AS) based on International Organization for Standardization (ISO) NSAPs.

A single routing domain consisting of ISO NSAP devices are considered to be CLNS islands. CLNS islands are connected together by VPNs.

You can configure BGP to exchange ISO CLNS routes between PE routers connecting various CLNS islands in a VPN using multiprotocol BGP extensions. These extensions are the ISO VPN NLRIs.

Each CLNS network island is treated as a separate VPN routing and forwarding instance (VRF) instance on the PE router.

You can configure CLNS on the global level, group level, and neighbor level.

### Example: Configuring BGP for CLNS VPNs

This example shows how to create a BGP group for CLNS VPNs, define the BGP peer neighbor address for the group, and define the family.

- [Requirements on page 570](#)
- [Overview on page 570](#)

- [Configuration on page 570](#)
- [Verification on page 570](#)

---

## Requirements

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

---

## Overview

In this example, you create the BGP group called pedge-pegde, define the BGP peer neighbor address for the group as 10.255.245.215, and define the BGP family.

---

## 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.

```
set protocols bgp group pedge-pegde neighbor 10.255.245.213
set protocols bgp family iso-vpn unicast
```

### 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 BGP for CLNS VPNs:

1. Configure the BGP group and define the BGP peer neighbor address.

```
[edit protocols bgp]
user@host# set group pedge-pegde neighbor 10.255.245.213
```

2. Define the family.

```
[edit protocols bgp]
user@host# set family iso-vpn unicast
```

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

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

---

## Verification

Confirm that the configuration is working properly.

### Verifying the Neighbor Status

**Purpose** Display information about the BGP peer.

**Action** From operational mode, run the **show bgp neighbor 10.255.245.213** command. Look for **iso-vpn-unicast** in the output.

```
user@host> show bgp neighbor 10.255.245.213
```

```

Peer: 10.255.245.213+179 AS 200 Local: 10.255.245.214+3770 AS 100
Type: External State: Established Flags: <ImportEval Sync>
Last State: OpenConfirm Last Event: RecvKeepAlive
Last Error: None
Options: <Multihop Preference LocalAddress HoldTime AddressFamily PeerAS
Rib-group Refresh>
Address families configured: iso-vpn-unicast
Local Address: 10.255.245.214 Holdtime: 90 Preference: 170
Number of flaps: 0
Peer ID: 10.255.245.213 Local ID: 10.255.245.214 Active Holdtime: 90
Keepalive Interval: 30 Peer index: 0
NLRI advertised by peer: iso-vpn-unicast
NLRI for this session: iso-vpn-unicast
Peer supports Refresh capability (2)
Table bgp.isovpn.0 Bit: 10000
RIB State: BGP restart is complete
RIB State: VPN restart is complete
Send state: in sync
Active prefixes: 3
Received prefixes: 3
Suppressed due to damping: 0
Advertised prefixes: 3
Table aaaa.iso.0
RIB State: BGP restart is complete
RIB State: VPN restart is complete
Send state: not advertising
Active prefixes: 3
Received prefixes: 3
Suppressed due to damping: 0
Last traffic (seconds): Received 6 Sent 5 Checked 5
Input messages: Total 1736 Updates 4 Refreshes 0 Octets 33385
Output messages: Total 1738 Updates 3 Refreshes 0 Octets 33305
Output Queue[0]: 0
Output Queue[1]: 0

```

## Enabling BGP to Carry CLNS Routes

Connectionless Network Service (CLNS) is a Layer 3 protocol similar to IP version 4 (IPv4). CLNS uses network service access points (NSAPs) to address end systems. This allows for a seamless autonomous system (AS) based on International Organization for Standardization (ISO) NSAPs.

A single routing domain consisting of ISO NSAP devices are considered to be CLNS islands. CLNS islands are connected together by VPNs.

You can configure BGP to exchange ISO CLNS routes between provider edge (PE) routers connecting various CLNS islands in a virtual private network (VPN) using multiprotocol BGP extensions. These extensions are the ISO VPN NLRIs.

To enable multiprotocol BGP (MP-BGP) to carry CLNS VPN NLRIs, include the **iso-vpn** statement:

```

iso-vpn {
  unicast {
    prefix-limit number;
    rib-group group-name;
  }
}

```

To limit the number of prefixes from a peer, include the **prefix-limit** statement. To specify a routing table group, include the **rib-group** statement.

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Each CLNS network island is treated as a separate VRF instance on the PE router.

You can configure CLNS on the global level, group level, and neighbor level.

For sample configurations, see the following sections:

- [Example: Enabling CLNS Between Two Routers on page 572](#)
- [Example: Configuring CLNS Within a VPN on page 574](#)

---

### Example: Enabling CLNS Between Two Routers

---

Configure CLNS between two routers through a route reflector:

On Router 1:

```
protocols {
  bgp {
    local-address 10.255.245.195;
    group pe-pe {
      type internal;
      neighbor 10.255.245.194 {
        family iso-vpn {
          unicast;
        }
      }
    }
  }
}
routing-instances {
  aaaa {
    instance-type vrf;
    interface fe-0/0/0.0;
    interface so-1/1/0.0;
    interface lo0.1;
    route-distinguisher 10.255.245.194:1;
    vrf-target target:11111:1;
    protocols {
      isis {
        export dist-bgp;
        no-ipv4-routing;
        no-ipv6-routing;
        clns-routing;
        interface all;
      }
    }
  }
}
On Router 2:
protocols {
  bgp {
```

```

group pe-pe {
    type internal;
    local-address 10.255.245.198;
    family route-target;
    neighbor 10.255.245.194 {
        family iso-vpn {
            unicast;
        }
    }
}
}
}
}
}
routing-instances {
    aaaa {
        instance-type vrf;
        interface lo0.1;
        interface so-0/1/2.0;
        interface so-0/1/3.0;
        route-distinguisher 10.255.245.194:1;
        vrf-target target:11111:1;
        routing-options {
            rib aaaa.iso.0 {
                static {
                    iso-route 47.0005.80ff.f800.0000.bbbb.1022/104 next-hop
                        47.0005.80ff.f800.0000.aaaa.1000.1921.6800.4196.00;
                }
            }
        }
    }
}
protocols {
    isis {
        export dist-bgp;
        no-ipv4-routing;
        no-ipv6-routing;
        clns-routing;
        interface all;
    }
}
}
}
}
On Route Reflector:
protocols {
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.245.194;
            family route-target;
            neighbor 10.255.245.195 {
                cluster 0.0.0.1;
            }
            neighbor 10.255.245.198 {
                cluster 0.0.0.1;
            }
        }
    }
}
}
}

```

### Example: Configuring CLNS Within a VPN

Configure CLNS on three PE routers within a VPN:

On PE Router 1:

```
protocols {
  mpls {
    interface all;
  }
  bgp {
    group asbr {
      type external;
      local-address 10.245.245.3;
      neighbor 10.245.245.1 {
        multihop;
        family iso-vpn {
          unicast;
        }
        peer-as 200;
      }
    }
  }
}
routing-instances {
  aaaa {
    instance-type vrf;
    interface lo0.1;
    interface t1-3/0/0.0;
    interface fe-5/0/1.0;
    route-distinguisher 10.245.245.1:1;
    vrf-target target:11111:1;
    protocols {
      isis {
        export dist-bgp;
        no-ipv4-routing;
        no-ipv6-routing;
        clns-routing;
        interface all;
      }
    }
  }
}
```

On PE Router 2:

```
protocols {
  bgp {
    group asbr {
      type external;
      multihop;
      family iso-vpn {
        unicast;
      }
      neighbor 10.245.245.2 {
        peer-as 300;
      }
      neighbor 10.245.245.3 {
```

```

        peer-as 100;
    }
}
}
routing-instances {
  aaaa {
    instance-type vrf;
    interface lo0.1;
    route-distinguisher 10.245.245.1:1;
    vrf-target target:11111:1;
  }
}
On PE Router 3:
protocols {
  bgp {
    group asbr {
      type external;
      multihop;
      local-address 10.245.245.2;
      neighbor 10.245.245.1 {
        family iso-vpn {
          unicast;
        }
        peer-as 200;
      }
    }
  }
}
routing-instances {
  aaaa {
    instance-type vrf;
    interface lo0.1;
    interface fe-0/0/1.0;
    interface t1-3/0/0.0;
    route-distinguisher 10.245.245.1:1;
    vrf-target target:11111:1;
    protocols {
      isis {
        export dist-bgp;
        no-ipv6-routing;
        clns-routing;
        interface all;
      }
    }
  }
}
}

```

**Related  
Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)



# BGP Monitoring Configuration

- [Configuring BGP Monitoring Protocol Version 3 on page 577](#)
- [Example: Configuring BGP Monitoring Protocol on page 580](#)
- [Example: Configuring BGP Trace Operations on page 583](#)
- [Tracing BMP Operations on page 589](#)

## Configuring BGP Monitoring Protocol Version 3

---

BGP Monitoring Protocol (BMP) allows the Junos OS to send the BGP route information from the router to a monitoring application on a separate device. The monitoring application is called the BMP monitoring station or BMP station. To deploy BMP in your network, you need to configure BMP on each router and you also need to configure at least one BMP station. This procedure describes how to configure BMP on a router.

You can specify these settings for all BMP stations by configuring the statements described here at the **[edit routing-options bmp]** hierarchy level. You can also configure settings for specific BMP stations by configuring these statements at the **[edit routing-options bmp station *station-name*]** hierarchy level.

The following procedure describes how to configure BMP version 3 on the router:

1. Specify the name or address for the BMP monitoring station by configuring the **station-address** statement. You can specify one or the other but not both. The address must be a valid IPv4 or IPv6 address.

**station-address** (*station-address* | *station-name*);

2. Specify the authentication algorithm used to encrypt authentication between the BMP-enabled router and the BMP station using the **authentication-algorithm** statement.

**authentication-algorithm** *algorithm*;

You can specify one of the following types of authentication algorithms:

- **aes-128-cmac-96**—Cipher-based message authentication code (AES128, 96 bits).
- **hmac-sha-1-96**—Hash-based message authentication code (SHA1, 96 bits).
- **md5**—Message digest 5.

3. Specify an MD5 authentication key (password) using the **authentication-key** statement.

```
authentication-key key;
```

4. Specify the authentication key chain using the **authentication-key-chain** statement.

```
authentication-key-chain key-chain;
```

The authentication key chain itself needs to be configured at the **[edit security authentication-key-chains key-chain]** hierarchy level. For a detailed example, see [“Example: Configuring Route Authentication for BGP” on page 450](#).

5. Specify how to handle a BMP station flap by configuring the **hold-down** statement. A flap is when the TCP session unexpectedly switches from established to non-established. The BMP station can be prevented from attempting to reconnect to the device for a specified period of time.

```
hold-down {  
    seconds;  
    flaps number;  
    period seconds;  
}
```

You can specify the following options for the **hold-down** statement:

- **seconds**—Specify the time in seconds to wait before allowing the BMP station to reconnect to the device.
  - **flaps number**—Specify the number of BMP station flaps allowed before terminating the connection to the BMP station and triggering the hold down timer.
  - **period seconds**—Specify the time in seconds between BMP station flaps before terminating the connection to the BMP station and triggering the hold down timer.
6. (Optional) Specify an initiation message to be sent to the BMP station using the **initiation-message** statement. This statement allows you to provide some information to the BMP station system administrator (for example, a contact phone number).

```
initiation-message text;
```

7. Specify the connection mode for the connection between the BMP-enabled router and the BMP station using the **connection-mode** statement. The connection mode can be **active** or **passive**:

- **active**—BMP initiates the connection to the BMP station. If you configure active mode, you must also configure a station port using the **station-port** statement. However, you must not configure a local port (active mode).
- **passive**—BMP does not initiate a connection the BMP station. However, it does listen for a connection request from active BMP stations and will connect if a station is available. If you configure passive mode, you must not configure a station port. However, you must configure a local port using the **local-port** statement (passive mode).

```
connection-mode (active | passive);
```

8. Specify the port number for the BMP monitoring station by configuring the **station-port** statement. See also [connection-mode](#).

```
station-port port;
```

9. Specify the listening port for the BMP station connection using the **local-port** statement. See also **connection-mode**.

If you change the local port, the BMP station connection flaps when you commit the configuration.

#### **local-port**

10. (Optional) Specify the IPv4 or IPv6 address for the BMP connection on the device using the **local-address** statement. For both active and passive connections, configure a loopback local address. This provides a consistent local endpoint, is useful for debugging, and assures greater reliability for the BMP connection since it is not tied to a single router interface.

For passive mode, specifying a local address is required. It also provides some security against a malicious BMP connection. For active mode, we also recommend configuring a local address to help ensure reliability.

If you change the local address, the BMP station connection flaps when you commit the configuration.

#### **local-address** *address*;

11. BMP monitoring is enabled by default. You can explicitly enable BMP monitoring or disable it. You can also selectively enable or disable BMP monitoring at various hierarchy levels (for example, [**edit protocols bgp group group-name**] or [**edit protocols bgp group group-name neighbor address**]). If you disable BMP monitoring, withdrawal messages are sent for any previously advertised routes. These are followed by a down message. If you enable BMP monitoring, an up message is sent first and then the route advertisements follow.

#### **monitor** (enable | disable);

12. Specify the dispatch priority for BMP by configuring the **priority** statement. The dispatch priority controls the frequency with which the device is able to forward BMP messages to BMP stations. You can configure the dispatch priority as either **high**, **medium**, or **low**.

#### **priority** (high | medium | low);

13. Specify whether BMP should send pre-policy route monitoring messages, post-policy route monitoring messages, both types of messages, or none at all. The pre-policy can be configured to exclude routes that are non-feasible for the decision process (for example, a route loop) by including the **non-feasible** option for the **pre-policy** statement. This represents the view of the BGP routes before running the import policy.

The post-policy can be configured to exclude routes that are not eligible for the decision process (for example, protocol nexthop not resolved) by including the **exclude-non-eligible** option for the **post-policy** statement. This represents the view of the BGP routes after running the import policy. If the import policy has rejected the BGP route, the route does not exist in the post policy view.

You can explicitly disable route monitoring by specifying the **none** option for the **route-monitoring** statement. This is the default behavior.

```
route-monitoring {
  none;
  post-policy {
```

```
        exclude-non-eligible;
    }
    pre-policy {
        exclude-non-feasible;
    }
}
```

14. Configure how often statistics messages are sent to the BMP monitoring station by specifying the number of seconds between message transmissions using **statistics-timeout** statement. If you configure a value of 0, no statistics messages are sent.

```
statistics-timeout seconds;
```

#### Related Documentation

- [Example: Configuring Route Authentication for BGP on page 450](#)

---

## Example: Configuring BGP Monitoring Protocol

- [Understanding the BGP Monitoring Protocol on page 580](#)
- [Example: Configuring the BGP Monitoring Protocol on page 580](#)

### Understanding the BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) is a protocol to allow a monitoring station to receive routes from a BGP-enabled device. The monitoring station receives all routes, not just the active routes. BMP uses route monitoring messages (which are essentially encapsulated BGP update messages) and a few other message types for statistics and state changes. All messages flow from the router to the monitoring station.

The data is collected from the **Adjacency-RIB-In** routing tables. The **Adjacency-RIB-In** tables are the pre-policy tables, meaning that the routes in these tables have not been filtered or modified by routing policies.



**NOTE:** The Local-RIB tables are the post-policy tables.

---

### Example: Configuring the BGP Monitoring Protocol

This example shows how to enable the BGP Monitoring Protocol (BMP). The Junos OS implementation of BMP is based on Internet draft draft-scurder-bmp-01.txt, *BGP Monitoring Protocol*.

- [Requirements on page 581](#)
- [Overview on page 581](#)
- [Configuration on page 581](#)
- [Verification on page 582](#)

## Requirements

- Configure the router interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP and routing policies.
- Configure a monitoring station to listen on a particular TCP port.

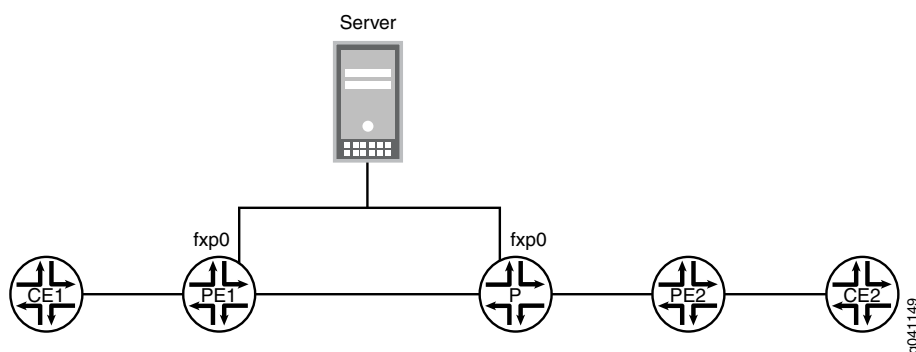
## Overview

To configure the monitoring station to which BMP data is sent, you must configure both the **station-address** and **station-port** statements. For the station address, you can specify either the IP address or the name of the monitoring station. For **name**, specify a valid URL. For the station port, specify a TCP port. BMP operates over TCP. The monitoring station is configured to listen on a particular TCP port, and the router is configured to establish an active connection to that port and to send messages on that TCP connection. You configure BMP in the default routing instance only. However, BMP applies to routes in the default routing instance and to routes in other routing instances.

You can optionally specify how often to send data to the monitoring station. The default is 1 hour. To modify this interval, include the **statistics-timeout seconds** statement. For **seconds**, you can specify a value from 15 through 65,535. By default, the routing device stops collecting BMP data when it exceeds a threshold of 10 MB. You can modify the value of this threshold by including the **memory-limit bytes** statement. For **bytes**, specify a value from 1,048,576 to 52,428,800. If the routing device stops collecting BMP data after exceeding the configured memory threshold, the router waits 10 minutes before attempting to resume the BMP session.

Figure 68 on page 581 shows a sample topology. In this example, BMP is configured on Router PE1. The server address is 192.168.64.180. The listening TCP port on the server is port 11019.

Figure 68: BMP Topology



## 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.

```
set routing-options bmp station-address 192.168.64.180
set routing-options bmp station-port 11019
```

**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 BMP:

1. Configure the receiving station address.

```
[edit routing-options]
user@PE1# set bmp station-address 192.168.64.180
```

2. Configure the receiving station port.

```
[edit routing-options]
user@PE1# set bmp station-port 11019
```

### Results

From configuration mode, confirm your configuration by entering the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show routing-options
bmp {
  station-address 192.168.64.180;
  station-port 11019;
}
```

---

### Verification

#### Verifying That BMP is Operating

**Purpose** Run the **show bgp bmp** command to display a set of statistics and the current BMP session state on the router.

**Action**

```
user@PE1> show bgp bmp
BMP station address/port: 192.168.64.180+11019
BMP session state: DOWN
Memory consumed by BMP: 0
Statistics timeout: 15
Memory limit: 10485760
Memory connect retry timeout: 600
```

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Example: Configuring BGP Trace Operations

- [Understanding Trace Operations for BGP Protocol Traffic on page 583](#)
- [Example: Viewing BGP Trace Files on Logical Systems on page 584](#)

### Understanding Trace Operations for BGP Protocol Traffic

You can trace various BGP protocol traffic to help you debug BGP protocol issues. To trace BGP protocol traffic, include the **traceoptions** statement at the **[edit protocols bgp]** hierarchy level. For routing instances, include the **traceoptions** statement at the **[edit routing-instances routing-instance-name protocols bgp]** hierarchy level.

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

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

- **4byte-as**—4-byte AS events.
- **bfd**—BFD protocol events.
- **damping**—Damping operations.
- **graceful-restart**—Graceful restart events.
- **keepalive**—BGP keepalive messages.
- **nsr-synchronization**—Nonstop active routing synchronization events.
- **open**—BGP open packets. These packets are sent between peers when they are establishing a connection.
- **packets**—All BGP protocol packets.
- **refresh**—BGP refresh packets.
- **update**—BGP update packets. These packets provide routing updates to BGP systems.

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 BGP protocol using the **traceoptions flag** statement included at the **[edit protocols bgp]** 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

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

- **detail**—Detailed trace information.
- **filter**—Filter trace information. Applies only to **route** and **damping** tracing flags.
- **receive**—Packets being received.
- **send**—Packets being transmitted.



**NOTE:** Use the **all** trace flag and the **detail** flag modifier with caution because these might cause the CPU to become very busy.

---



**NOTE:** If you only enable the **update** flag, received keepalive messages do not generate a trace message.

---

You can filter trace statements and display only the statement information that passes through the filter by specifying the **filter** flag modifier. The **filter** modifier is only supported for the **route** and **damping** tracing flags.

The **match-on** statement specifies filter matches based on prefixes. It is used to match on route filters.



**NOTE:** Per-neighbor trace filtering is not supported on a BGP per-neighbor level for **route** and **damping** flags. Trace option filtering support is on a peer group level.

---

## Example: Viewing BGP Trace Files on Logical Systems

This example shows how to list and view files that are stored on a logical system.

- [Requirements on page 584](#)
- [Overview on page 585](#)
- [Configuration on page 585](#)
- [Verification on page 589](#)

---

### Requirements

- You must have the **view** privilege for the logical system.
- Configure a network, such as the BGP network shown in “[Example: Configuring Internal BGP Peering Sessions on Logical Systems](#)” on page 68.

## Overview

Logical systems have their individual directory structure created in the `/var/logical-systems/logical-system-name` directory. It contains the following subdirectories:

- `/config`—Contains the active configuration specific to the logical system.
- `/log`—Contains system log and tracing files specific to the logical system.

To maintain backward compatibility for the log files with previous versions of Junos OS, a symbolic link (symlink) from the `/var/logs/logical-system-name` directory to the `/var/logical-systems/logical-system-name` directory is created when a logical system is configured.

- `/tmp`—Contains temporary files specific to the logical system.

The file system for each logical system enables logical system users to view trace logs and modify logical system files. Logical system administrators have full access to view and modify all files specific to the logical system.

Logical system users and administrators can save and load configuration files at the logical-system level using the **save** and **load** configuration mode commands. In addition, they can also issue the **show log**, **monitor**, and **file** operational mode commands at the logical-system level.

This example shows how to configure and view a BGP trace file on a logical system. The steps can be adapted to apply to trace operations for any Junos OS hierarchy level that supports trace operations.



**TIP:** To view a list of hierarchy levels that support tracing operations, enter the `help apropos traceoptions` command in configuration mode.

## Configuration

- [Configuring Trace Operations on page 586](#)
- [Viewing the Trace File on page 586](#)
- [Deactivating and Reactivating Trace Logging on page 588](#)
- [Results on page 589](#)

### 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.

```
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-log
set logical-systems A protocols bgp group internal-peers traceoptions file size 10k
set logical-systems A protocols bgp group internal-peers traceoptions file files 2
set logical-systems A protocols bgp group internal-peers traceoptions flag update detail
```

### Configuring Trace Operations

**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 trace operations:

1. Configure trace operations on the logical system.  
  

```
[edit logical-systems A protocols bgp group internal-peers]
user@host# set traceoptions file bgp-log
user@host# set traceoptions file size 10k
user@host# set traceoptions file files 2
user@host# set traceoptions flag update detail
```
2. If you are done configuring the device, commit the configuration.  
  

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

### Viewing the Trace File

**Step-by-Step Procedure** To view the trace file:

1. In operational mode on the main router, list the directories on the logical system.  
  

```
user@host> file list /var/logical-systems/A
/var/logical-systems/A:
config/
log/
tmp/
```
2. In operational mode on the main router, list the log files on the logical system.  
  

```
user@host> file list /var/logical-systems/A/log/
/var/logical-systems/A/log:
bgp-log
```
3. View the contents of the **bgp-log** file.  
  

```
user@host> file show /var/logical-systems/A/log/bgp-log
Aug 10 17:12:01 trace_on: Tracing to "/var/log/A/bgp-log" started
Aug 10 17:14:22.826182 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.163.6.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.826445 bgp_send: sending 21 bytes to 192.163.6.4 (Internal
AS 17)
Aug 10 17:14:22.826499
Aug 10 17:14:22.826499 BGP SEND 192.168.6.5+64965 -> 192.163.6.4+179
Aug 10 17:14:22.826559 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.826598 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
Aug 10 17:14:22.831756 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.168.40.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.831851 bgp_send: sending 21 bytes to 192.168.40.4 (Internal
AS 17)
Aug 10 17:14:22.831901
Aug 10 17:14:22.831901 BGP SEND 192.168.6.5+53889 -> 192.168.40.4+179
```

```
Aug 10 17:14:22.831959 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.831999 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
...
```

4. Filter the output of the log file.

```
user@host> file show /var/logical-systems/A/log/bgp-log | match "flags 0x40"
Aug 10 17:14:54.867460 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.867595 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.867650 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.867692 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.884529 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.884581 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.884628 BGP RECV flags 0x40 code NextHop(3): 192.168.6.4
Aug 10 17:14:54.884667 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.911377 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.911422 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.911466 BGP RECV flags 0x40 code NextHop(3): 192.168.40.4
Aug 10 17:14:54.911507 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.916008 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.916054 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.916100 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.916143 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.920304 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.920348 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.920393 BGP RECV flags 0x40 code NextHop(3): 10.0.0.10
Aug 10 17:14:54.920434 BGP RECV flags 0x40 code LocalPref(5): 100
```

5. View the tracing operations in real time.

```
user@host> clear bgp neighbor logical-system A
Cleared 2 connections
```



**CAUTION:** Clearing the BGP neighbor table is disruptive in a production environment.

6. Run the **monitor start** command with an optional **match** condition.

```
user@host> monitor start A/bgp-log | match 0.0.0.0/0
Aug 10 19:21:40.773467 BGP RECV          0.0.0.0/0
Aug 10 19:21:40.773685 bgp_rcv_nlrri: 0.0.0.0/0
Aug 10 19:21:40.773778 bgp_rcv_nlrri: 0.0.0.0/0 belongs to meshgroup
Aug 10 19:21:40.773832 bgp_rcv_nlrri: 0.0.0.0/0 qualified bnp->ribact 0x0
12afcb 0x0
```

7. Pause the **monitor** command by pressing Esc-Q.  
To unpause the output, press Esc-Q again.
8. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]
user@host> monitor stop
```

9. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, it appears in the configuration with the **inactive** tag. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

10. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

### *Deactivating and Reactivating Trace Logging*

#### **Step-by-Step Procedure**

To deactivate and reactivate the trace file:

1. When you are finished troubleshooting, consider deactivating trace logging to avoid an unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, the statement appears in the configuration with the **inactive** tag.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

2. To reactivate logging, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

### Results

From configuration mode, confirm your configuration by entering the **show logical-systems A protocols bgp group internal-peers** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems A protocols bgp group internal-peers
traceoptions {
  file bgp-log size 10k files 2;
  flag update detail;
}
```

### Verification

Confirm that the configuration is working properly.

#### Verifying That the Trace Log File Is Operating

**Purpose** Make sure that events are being written to the log file.

**Action** user@host:A> **show log bgp-log**  
 Aug 12 11:20:57 trace\_on: Tracing to "/var/log/A/bgp-log" started

**Related Documentation**

- [Understanding External BGP Peering Sessions on page 33](#)
- [BGP Configuration Overview on page 12](#)

## Tracing BMP Operations

You can trace BMP operations for all BMP stations by configuring the **traceoptions** statement at the **[edit routing-options bmp]** hierarchy level or for specific BMP stations at the **[edit routing-options bmp station station-name]** hierarchy level.

To trace BMP operations, complete the following steps:

1. Configure the **traceoptions** statement:

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

2. Specify the name of the file to receive the output of the tracing operation using the **file** option. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place BMP tracing output in the file **bmp-log**.
3. (Optional) Specify the maximum number of trace files using the **files** option. When a trace file named **trace-file.0** 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.

4. (Optional) Specify the maximum size of each trace file using the **size** option in kilobytes (KB), megabytes (MB), or gigabytes (GB). 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 also must specify a maximum number of trace files with the **files** option.
5. (Optional) You can specify that the log files are either **world-readable** (accessible to all users on the device) or **no-world-readable** (not accessible to all users on the device).
6. You can specify the following BMP-specific trace options using the **flag** statement:
  - **all**—Trace all BMP monitoring operations.
  - **down**—Down messages.
  - **error**—Error conditions.
  - **event**—Major events, session establishment, errors, and events.
  - **general**—General events.
  - **normal**—Normal events.
  - **packets**—All messages.
  - **policy**—Policy processing.
  - **route**—Routing information.
  - **route-monitoring**—Route monitoring messages.
  - **state**—State transitions.
  - **statistics**—Statistics messages.
  - **task**—Routing protocol task processing.
  - **timer**—Routing protocol timer processing.
  - **up**—Up messages.
  - **write**—Writing of messages.

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

- **detail**—Provide detailed trace information.
- **disable**—Disable the tracing flag.
- **receive**—Trace the packets being received.
- **send**—Trace the packets being transmitted.



**NOTE:** Use the **all** trace flag and the **detail** flag modifier with caution due to the increased computer processing power required.

---

- Related Documentation**
- [Configuring BGP Monitoring Protocol Version 3 on page 577](#)



## CHAPTER 15

# BGP Configuration Statements

- [\[edit protocols bgp\] Hierarchy Level on page 597](#)
- [accept-remote-nexthop on page 604](#)
- [accepted-prefix-limit on page 605](#)
- [add-path on page 607](#)
- [advertise-external on page 608](#)
- [advertise-from-main-vpn-tables on page 610](#)
- [advertise-inactive on page 611](#)
- [advertise-peer-as on page 612](#)
- [aggregate-label on page 613](#)
- [aigp on page 614](#)
- [aigp-originate on page 616](#)
- [algorithm \(BGP BFD Authentication\) on page 617](#)
- [allow on page 619](#)
- [as-override on page 620](#)
- [authentication \(BGP BFD Liveness Detection\) on page 621](#)
- [authentication-algorithm on page 623](#)
- [authentication-key \(Protocols BGP and BMP\) on page 624](#)
- [authentication-key-chain \(Protocols BGP and BMP\) on page 625](#)
- [auto-discovery-only on page 626](#)
- [bfd-liveness-detection \(Protocols BGP\) on page 627](#)
- [bgp on page 631](#)
- [bgp-error-tolerance \(Protocols BGP\) on page 631](#)
- [bgp-orf-cisco-mode on page 632](#)
- [bmp on page 634](#)
- [cluster on page 636](#)
- [connection-mode on page 637](#)
- [damping \(Protocols BGP\) on page 638](#)
- [description \(Protocols BGP\) on page 640](#)

- [detection-time \(BFD Liveness Detection\) on page 641](#)
- [disable \(Protocols BGP\) on page 642](#)
- [disable \(BGP Graceful Restart\) on page 643](#)
- [explicit-null \(Protocols BGP\) on page 644](#)
- [export \(Protocols BGP\) on page 646](#)
- [family \(Protocols BGP\) on page 647](#)
- [file \(Tracing for Origin AS Validation\) on page 651](#)
- [flag \(Tracing for Origin AS Validation\) on page 652](#)
- [flow on page 653](#)
- [graceful-restart \(Protocols BGP\) on page 654](#)
- [group \(Protocols BGP\) on page 655](#)
- [group \(Origin Validation for BGP\) on page 658](#)
- [hold-down on page 659](#)
- [hold-down-interval \(BGP BFD Liveness Detection\) on page 661](#)
- [hold-time \(Protocols BGP\) on page 663](#)
- [hold-time \(Origin Validation for BGP\) on page 665](#)
- [idle-after-switch-over on page 666](#)
- [import \(Protocols BGP\) on page 667](#)
- [include-mp-next-hop on page 669](#)
- [inet-mdt \(Signaling\) on page 670](#)
- [initiation-message on page 671](#)
- [ipsec-sa \(Protocols BGP\) on page 672](#)
- [iso-vpn on page 673](#)
- [keep on page 674](#)
- [key-chain \(BGP BFD Authentication\) on page 676](#)
- [labeled-unicast \(Protocols BGP\) on page 678](#)
- [local-address \(Protocols BGP\) on page 680](#)
- [local-address \(Protocols BMP\) on page 682](#)
- [local-address \(Origin Validation for BGP\) on page 683](#)
- [local-as on page 684](#)
- [local-interface \(IPv6\) on page 686](#)
- [local-port on page 687](#)
- [local-preference on page 688](#)
- [log-updown \(Protocols BGP\) on page 689](#)
- [logical-systems on page 690](#)
- [loops on page 691](#)
- [loose-check \(BGP BFD Authentication\) on page 693](#)

- [malformed-route-limit \(Protocols BGP\) on page 694](#)
- [malformed-update-log-interval \(Protocols BGP\) on page 695](#)
- [max-sessions \(Origin Validation for BGP\) on page 696](#)
- [maximum-length \(Origin Validation for BGP\) on page 697](#)
- [metric-out \(Protocols BGP\) on page 698](#)
- [minimum-interval \(BFD Liveness Detection\) on page 700](#)
- [minimum-interval \(transmit-interval\) on page 702](#)
- [minimum-receive-interval \(BFD Liveness Detection\) on page 704](#)
- [monitor \(Protocols BMP\) on page 705](#)
- [mtu-discovery on page 706](#)
- [multihop on page 708](#)
- [multipath \(Protocols BGP\) on page 710](#)
- [multiplier \(BFD Liveness Detection\) on page 711](#)
- [neighbor \(Protocols BGP\) on page 713](#)
- [no-adaptation \(BFD Liveness Detection\) on page 716](#)
- [no advertise-peer-as on page 717](#)
- [no-aggregator-id on page 718](#)
- [no-client-reflect on page 719](#)
- [no-malformed-route-limit \(Protocols BGP\) on page 720](#)
- [no-nexthop-change \(BGP multihop\) on page 721](#)
- [no-validate on page 723](#)
- [origin-autonomous-system \(Origin Validation for BGP\) on page 724](#)
- [out-delay on page 725](#)
- [outbound-route-filter on page 727](#)
- [passive \(Protocols BGP\) on page 728](#)
- [path-count on page 729](#)
- [path-selection on page 730](#)
- [peer-as \(Protocols BGP\) on page 732](#)
- [port \(Origin Validation for BGP\) on page 733](#)
- [pre-policy on page 734](#)
- [precision-timers on page 735](#)
- [preference \(Protocols BGP\) on page 736](#)
- [preference \(Origin Validation for BGP\) on page 737](#)
- [prefix-limit on page 738](#)
- [prefix-policy on page 740](#)
- [priority \(Protocols BMP\) on page 741](#)
- [post-policy on page 742](#)

- [protection \(Protocols BGP\) on page 742](#)
- [protection \(Protocols MPLS\) on page 743](#)
- [protocols on page 744](#)
- [receive \(Protocols BGP\) on page 746](#)
- [record \(Origin Validation for BGP\) on page 747](#)
- [record-lifetime \(Origin Validation for BGP\) on page 748](#)
- [refresh-time \(Origin Validation for BGP\) on page 749](#)
- [remove-private on page 750](#)
- [resolve-vpn on page 752](#)
- [restart-time \(BGP Graceful Restart\) on page 753](#)
- [rib \(Protocols BGP\) on page 754](#)
- [rib-group \(Protocols BGP\) on page 755](#)
- [route-monitoring on page 756](#)
- [route-target \(Protocols BGP\) on page 757](#)
- [routing-instances \(Multiple Routing Entities\) on page 758](#)
- [send \(Logical Systems Add-Path\) on page 759](#)
- [session \(Origin Validation for BGP\) on page 760](#)
- [session-mode on page 761](#)
- [stale-routes-time on page 762](#)
- [static \(Origin Validation for BGP\) on page 763](#)
- [station on page 764](#)
- [station-address on page 765](#)
- [station-port on page 766](#)
- [statistics-timeout on page 767](#)
- [tcp-aggressive-transmission on page 768](#)
- [tcp-mss \(Protocols BGP\) on page 769](#)
- [threshold \(detection-time\) on page 770](#)
- [threshold \(transmit-interval\) on page 772](#)
- [topology \(Protocols BGP\) on page 774](#)
- [traceoptions \(Protocols BGP\) on page 776](#)
- [traceoptions \(Protocols BMP\) on page 779](#)
- [traceoptions \(Origin Validation for BGP\) on page 781](#)
- [traffic-statistics \(Protocols BGP\) on page 782](#)
- [transmit-interval \(BFD Liveness Detection\) on page 783](#)
- [ttl \(Protocols BGP\) on page 785](#)
- [type \(Protocols BGP\) on page 787](#)
- [validation \(Origin Validation for BGP\) on page 788](#)

- [validation-state](#) (Origin Validation for BGP) on page 789
- [version](#) (BFD Liveness Detection) on page 790
- [vpn-apply-export](#) on page 791

## [edit protocols bgp] Hierarchy Level

Several statements in the **[edit protocols mpls]** hierarchy are valid at numerous locations within it. To make the complete hierarchy easier to read, the repeated statements are listed in “[Common BGP Family Options](#)” on page 597 and that section is referenced at the appropriate locations in “[Complete \[edit protocols bgp\] Hierarchy](#)” on page 597.

- [Common BGP Family Options](#) on page 597
- [Complete \[edit protocols bgp\] Hierarchy](#) on page 597

## Common BGP Family Options

This section lists statements that are valid at the following hierarchy levels, and is referenced at those levels in “[Complete \[edit protocols bgp\] Hierarchy](#)” on page 597 instead of the statements being repeated.

- **[edit protocols bgp family inet (any | flow | labeled-unicast | multicast | unicast)]**
- **[edit protocols bgp family inet6 (any | labeled-unicast | multicast | unicast)]**
- **[edit protocols bgp family (evpn | inet-mdt | inet-mvpn | inet6-mvpn | l2vpn) signaling]**
- **[edit protocols bgp family inet-vpn (any | flow | multicast | unicast)]**
- **[edit protocols bgp family inet6-vpn (any | multicast | unicast)]**
- **[edit protocols bgp family iso-vpn unicast]**

The common BGP family options are as follows:

```
accepted-prefix-limit {
    maximum number;
    teardown <percentage> <idle-timeout (forever | minutes)>;
}
damping;
loops number;
prefix-limit {
    maximum number;
    teardown <percentage> <idle-timeout (forever | minutes)>;
}
rib-group group-name;
topology name {
    community {
        target identifier;
    }
}
```

## Complete [edit protocols bgp] Hierarchy

The statement hierarchy listed in this section can also be included at the **[edit logical-systems *logical-system-name*]** hierarchy level.

```

protocols {
  bgp {
    disable;
    accept-remote-nexthop;
    advertise-external <conditional>;
    advertise-from-main-vpn-tables;
    advertise-inactive;
    (advertise-peer-as | no-advertise-peer-as);
    authentication-algorithm (aes-128-cmac-96 | hmac-sha-1-96 | md5);
    authentication-key key;
    authentication-key-chain key-chain;
    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;
      }
      holddown-interval milliseconds;
      minimum-interval milliseconds;
      minimum-receive-interval milliseconds;
      multiplier number;
      no-adaptation;
      session-mode (automatic | multihop | single-hop);
      transmit-interval {
        minimum-interval milliseconds;
        threshold milliseconds;
      }
      version (1 | automatic);
    }
    bmp {
      monitor (disable | enable);
      route-monitoring {
        none;
        post-policy {
          exclude-non-eligible;
        }
        pre-policy {
          exclude-non-feasible;
        }
      }
    }
    cluster cluster-identifier;
    damping;
    description text-description;
    export [ policy-names ];
    family family-name {
      ... the family subhierarchies appear after the main [edit protocols bgp] hierarchy ...
    }
    unconfigured-peer-graceful-restart;
    graceful-restart {
      disable;
      restart-time seconds;
    }
  }
}

```

```

    stale-routes-time seconds;
}
group group-name {
    ... the group subhierarchy appears after the main [edit protocols bgp] hierarchy ...
}
hold-time seconds;
idle-after-switch-over (seconds | forever);
import [ policy-names ];
include-mp-next-hop;
ipsec-sa ipsec-sa;
keep (all | none);
local-address address;
local-as autonomous-system <loops number> <alias> <private>;
local-interface interface-name;
local-preference local-preference;
log-updown;
metric-out (metric | igp (delay-med-update | offset) | minimum-igp offset);
mtu-discovery;
multihop {
    no-nexthop-change;
    ttl ttl-value;
}
no-aggregator-id;
no-client-reflect;
out-delay seconds;
outbound-route-filter {
    bgp-orf-cisco-mode;
    prefix-based {
        accept {
            inet;
            inet6;
        }
    }
}
passive;
path-selection {
    always-compare-med;
    as-path-ignore;
    cisco-non-deterministic;
    external-router-id;
    med-plus-igp {
        igp-multiplier number;
        med-multiplier number;
    }
}
peer-as autonomous-system;
preference preference;
remove-private;
tcp-mss segment-size;
traceoptions {
    file filename <files number> <size maximum-file-size> <world-readable |
        no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
tcp-aggressive-transmission;
vpn-apply-export;

```

```

}

bgp {
  family inet {
    (any | multicast) {
      ... statements in Common BGP Family Options on page 597 ...
    }
    flow {
      ... statements in Common BGP Family Options on page 597 PLUS ...
      no-validate [ validation-procedure-names ];
    }
    labeled-unicast {
      ... statements in Common BGP Family Options on page 597 PLUS ...
      add-path {
        receive;
        send {
          path-count number;
          prefix-policy [ policy-names ];
        }
      }
      aggregate-label {
        community community-name;
      }
      aigp [disable];
      explicit-null connected-only;
      per-group-label;
      per-prefix-label;
      protection;
      resolve-vpn;
      rib (inet.3 | inet6.3);
      traffic-statistics {
        file filename <files number> <size maximum-file-size> <world-readable |
          no-world-readable>;
        interval seconds;
      }
    }
  }
  unicast {
    ... statements in Common BGP Family Options on page 597 PLUS ...
    add-path {
      receive;
      send {
        path-count number;
        prefix-policy [ policy-names ];
      }
    }
    topology name {
      community target identifier;
    }
  }
}

bgp {
  family inet6 {
    (any | multicast) {
      ... statements in Common BGP Family Options on page 597 ...
    }
  }
}

```

```

}
labeled-unicast {
... statements in Common BGP Family Options on page 597 PLUS ...
add-path {
    receive;
    send {
        path-count number;
        prefix-policy [ policy-names ];
    }
}
aggregate-label {
    community community-name;
}
aigp [disable];
explicit-null;
per-group-label;
protection;
traffic-statistics {
    file filename <files number> <size maximum-file-size> <world-readable |
        no-world-readable>;
    interval seconds;
}
}
unicast {
... statements in Common BGP Family Options on page 597 PLUS ...
    topology name {
        community target identifier;
    }
}
}
}

bgp {
    family (evpn | inet-mdt | inet-mvpn | inet6-mvpn | l2vpn) {
        auto-discovery-only; # for l2vpn
        signaling {
... statements in Common BGP Family Options on page 597 ...
        }
    }
}

bgp {
    family inet-vpn {
        (any | multicast | unicast) {
... statements in Common BGP Family Options on page 597 PLUS ...
            aggregate-label <community community-name>;
        }
        flow {
... statements in Common BGP Family Options on page 597 ...
        }
    }
}

bgp {
    family inet6-vpn {
        (any | multicast | unicast) {

```

```

        ... statements in Common BGP Family Options on page 597 PLUS ...
        aggregate-label <community community-name>;
    }
}

bgp {
    family iso-vpn {
        unicast {
            ... statements in Common BGP Family Options on page 597 PLUS ...
            aggregate-label <community community-name>;
        }
    }
}

bgp {
    family route-target {
        accepted-prefix-limit {
            maximum number;
            teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        advertise-default;
        external-paths number;
        prefix-limit {
            maximum number;
            teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        proxy-generate <route-target-policy route-target-policy-name>;
    }
}

bgp {
    group group-name {
        ... same statements as at the [edit protocols bgp] hierarchy level PLUS ...
        allow [ all ip-prefix</prefix-length> ];
        as-override;
        multipath <multiple-as>;
        neighbor address {
            ... the neighbor subhierarchy appears after the main [edit protocols bgp group
                group-name] hierarchy ...
        }
        type (external | internal);
        ... BUT NOT ...
        disable; # NOT valid at this level
        group group-name { ... } # NOT valid at this level
        path-selection { ... } # NOT valid at this level
    }

    group group-name {
        neighbor address {
            ... same statements as at the [edit protocols bgp] hierarchy level PLUS ...
            as-override;
            multipath <multiple-as>;
            ... BUT NOT ...
            disable; # NOT valid at this level
            group group-name { ... } # NOT valid at this level
        }
    }
}

```

```
neighbor address { ... } # NOT valid at this level
path-selection { ... } # NOT valid at this level
}
}
}
}
```

- Related Documentation**
- *Notational Conventions Used in Junos OS Configuration Hierarchies*
  - *[edit protocols] Hierarchy Level*

## accept-remote-nexthop

<b>Syntax</b>	<code>accept-remote-nexthop;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp group <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<p>Statement 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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify that a single-hop EBGp peer accepts a remote next hop with which it does not share a common subnet. Configure a separate import policy on the EBGp peer to specify the remote next hop. You cannot configure <b>multihop</b> and <b>accept-remote-nexthop</b> statements for the same EBGp peer.</p> <p>For Junos OS Release 13.3 and later releases, specify that a multihop EBGp peer accepts a remote next hop with which it does not share a common subnet. This allows working around current resolver limitations to realize multipath forwarding in recursive next-hop resolution scenarios.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Single-Hop EBGp Peers to Accept Remote Next Hops on page 371</a></li> <li>• <a href="#">Understanding Route Advertisement on page 233</a></li> <li>• <a href="#">multipath on page 710</a></li> </ul>

## accepted-prefix-limit

**Syntax**    `accepted-prefix-limit {  
                  maximum number;  
                  teardown <percentage-threshold> idle-timeout (forever | minutes);  
                  }`

**Hierarchy Level**    [edit logical-systems *logical-system-name* protocols bgp family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* protocols bgp family route-target],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family route-target],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family route-target],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family route-target],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family route-target],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family route-target],  
[edit protocols bgp family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit protocols bgp family route-target],  
[edit protocols bgp group *group-name* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit protocols bgp group *group-name* family route-target],  
[edit protocols bgp group *group-name* neighbor *address* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit protocols bgp group *group-name* neighbor *address* family route-target],  
[edit routing-instances *routing-instance-name* protocols bgp family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit routing-instances *routing-instance-name* protocols bgp family route-target],  
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* family route-target],  
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family (inet | inet6) (any | flow | labeled-unicast | multicast | unicast)],  
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family route-target]

**Release Information**    Statement introduced in Junos OS Release 9.2.  
Statement introduced in Junos OS Release 9.2 for EX Series switches.

<b>Description</b>	<p>Configure a limit to the number of prefixes that can be accepted on a BGP peer session. When that limit is exceeded, a system log message is sent. You can optionally specify to reset the BGP session when the number of accepted prefixes exceeds the specified limit.</p> <p>This statement provides the ability to log a message, reset the BGP session, or do both when the number of prefixes received from the peer and accepted by policy exceeds a preset limit. This functionality is identical to the <b>prefix-limit</b> functionality except that it operates against accepted prefixes rather than received prefixes.</p>
<b>Options</b>	<p><b>maximum <i>number</i></b>—When you set the maximum number of prefixes, a message is logged when that number is exceeded.</p> <p><b>Range:</b> 1 through 4,294,967,295 (<math>2^{32} - 1</math>)</p> <p><b>teardown <i>&lt;percentage&gt;</i></b>—(Optional) If you include the <b>teardown</b> statement, the session is torn down when the maximum number of prefixes is reached. If you specify a percentage, messages are logged when the number of prefixes exceeds that percentage. After the session is torn down, it is reestablished in a short time unless you include the <b>idle-timeout</b> statement. Then the session can be kept down for a specified amount of time, or forever. If you specify <b>forever</b>, the session is reestablished only after you issue a <b>clear bgp neighbor</b> command.</p> <p><b>Range:</b> 1 through 100</p> <p><b>idle-timeout (<i>forever</i>   <i>timeout-in-minutes</i>)</b>—(Optional) If you include the <b>idle-timeout</b> statement, the session is torn down for a specified amount of time, or forever. If you specify a period of time, the session is allowed to reestablish after this timeout period. If you specify <b>forever</b>, the session is reestablished only after you intervene with a <b>clear bgp neighbor</b> command.</p> <p><b>Range:</b> 1 through 2400</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">prefix-limit on page 738</a></li><li>• <a href="#">Understanding Multiprotocol BGP on page 537</a></li></ul>

## add-path

---

<b>Syntax</b>	<pre> add-path {   receive;   send {     path-count <i>number</i>;     prefix-policy [ <i>policy-names</i> ];   } } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor address family <i>family</i>],</p> <p>[edit protocols bgp group <i>group-name</i> family <i>family</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor address family <i>family</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 11.3.
<b>Description</b>	<p>Enable advertisement of multiple paths to a destination, instead of advertising only the active path.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Advertising Multiple BGP Paths to a Destination on page 392</a></li> </ul>

## advertise-external

<b>Syntax</b>	<code>advertise-external {conditional};</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit protocols bgp group <i>group-name</i>],          [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-address</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.3.</p> <p>Statement introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify BGP to advertise the best external route into an IBGP mesh group, a route reflector cluster, or an AS confederation even if the best route is an internal route.</p> <p>In general, deployed BGP implementations do not advertise the external route with the highest local preference value to internal peers unless it is the best route. Although this behavior was required by an earlier version of the BGP version 4 specification, RFC 1771, it was typically not followed in order to minimize the amount of advertised information and to prevent routing loops. However, there are scenarios in which advertising the best external route is beneficial, in particular, situations that can result in IBGP route oscillation.</p> <p>The <b>advertise-external</b> statement is supported at both the group and neighbor level. If you configure the statement at the neighbor level, you must configure it for all neighbors in a group. Otherwise, the group is automatically split into different groups.</p> <p>In a confederation, when advertising a route to a confederation border router, any route from a different confederation sub-AS is considered external. When configuring the <b>advertise-external</b> statement for an AS confederation, it is recommended that EBGp peers belonging to different autonomous systems are configured in a separate EBGp peer group. This ensures consistency while BGP sends the best external route to peers in the configured peer group.</p> <p>To configure the <b>advertise-external</b> statement on a route reflector, you must disable intracluster reflection with the <b>no-client-reflect</b> statement.</p> <p>When a routing device is configured as a route reflector for a cluster, a route advertised by the route reflector is considered internal if it is received from an internal peer with the same cluster identifier or if both peers have no cluster identifier configured. A route received from an internal peer that belongs to another cluster, that is, with a different cluster identifier, is considered external.</p> <p>The <b>conditional</b> option causes BGP to advertise the external route only if the route selection process reaches the point where the multiple exit discriminator (MED) metric</p>

is evaluated. As a result, an external route with an AS path longer than that of the active path is not advertised.

Junos OS also provides support for configuring a BGP export policy that matches on the state of an advertised route. You can match on either active or inactive routes.

**Default** BGP does not advertise the external route with the highest local preference value to internal peers unless it is the best route.


**Options** **conditional**—(Optional) Advertise the best external path only if the route selection process reaches the point at which the multiple exit discriminator (MED) metric is evaluated. The **conditional** option restricts advertisement to when the best external path and the active path are equal until the MED step of the route selection process. This implies that external routes with a longer AS path length than the active path, for instance, are not advertised. The criteria used for selecting the best external path is the same whether or not the **conditional** option is configured.

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**

- [Example: Configuring BGP to Advertise the Best External Route to Internal Peers on page 304](#)
- [advertise-inactive on page 611](#)

## advertise-from-main-vpn-tables

<b>Syntax</b>	advertise-from-main-vpn-tables;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp],
<b>Release Information</b>	Statement introduced in Junos OS Release 12.3.
<b>Description</b>	<p>Advertise VPN routes from the main VPN tables in the master routing instance (for example, <i>bgp.l3vpn.0</i>, <i>bgp.mvpn.0</i>) instead of advertising VPN routes from the tables in the VPN routing instances (for example, <i>instance-name.inet.0</i>, <i>instance-name.mvpn.0</i>). Enable nonstop active routing (NSR) support for BGP multicast VPN (MVPN).</p> <p>When this statement is enabled, before advertising a route for a VPN prefix, the path selection algorithm is run on all routes (local and received) that have the same route distinguisher (RD).</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> <b>NOTE:</b> Adding or removing this statement causes all BGP sessions that have VPN address families to be removed and then added again. On the other hand, having this statement in the configuration prevents BGP sessions from going down when route reflector (RR) or autonomous system border router (ASBR) functionality is enabled or disabled on a routing device that has VPN address families configured.</p> </div>
<b>Default</b>	If you do not include this statement, VPN routes are advertised from the tables in the VPN routing instances.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding Junos OS Routing Tables</i></li> <li>• <i>Types of VPNs</i></li> </ul>

## advertise-inactive

<b>Syntax</b>	advertise-inactive;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit protocols bgp],          [edit protocols bgp <b>group</b> <i>group-name</i>],          [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Configure the routing table to export to BGP the best route learned by BGP even if Junos OS did not select this route to be an active route.</p> <p>One way to achieve multivendor compatibility is to include the <b>advertise-inactive</b> statement in the external BGP (EBGP) configuration. By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. The <b>advertise-inactive</b> statement causes Junos OS to advertise the best BGP route that is inactive because of IGP preference. When you use the <b>advertise-inactive</b> statement, the Junos OS device uses, for example, the OSPF route for forwarding, and the other vendor's device uses the EBGP route for forwarding. However, from the perspective of an EBGP peer in a neighboring AS, both vendors' devices appear to behave the same way.</p>
<b>Default</b>	By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Setting BGP to Advertise Inactive Routes on page 298</a></li> <li>• <a href="#">Example: Configuring the Preference Value for BGP Routes on page 252</a></li> <li>• <a href="#">Example: Configuring BGP Route Preference (Administrative Distance) on page 250</a></li> </ul>

- [advertise-external on page 608](#)

## advertise-peer-as

<b>Syntax</b>	advertise-peer-as;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit protocols bgp],          [edit protocols bgp <b>group</b> <i>group-name</i>],          [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Disable the default behavior of suppressing AS routes.</p> <p>If you include the <b>advertise-peer-as</b> statement in the configuration, BGP advertises routes learned from one external BGP (EBGP) peer back to another EBGP peer in the same autonomous system (AS) but not back to the originating peer.</p> <p>Another way to disable the route suppression default behavior is with the <b>as-override</b> statement. If you include both the <b>as-override</b> and <b>no-advertise-peer-as</b> statements in the configuration, the <b>no-advertise-peer-as</b> statement is ignored.</p>
<b>Default</b>	By default, Junos OS does not advertise the routes learned from one EBGP peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGP peers that are in the same AS as the originating peer, regardless of the routing instance.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Disabling Suppression of Route Advertisements on page 200</a></li> <li>• <a href="#">Example: Configuring a Layer 3 VPN with Route Reflection and AS Override on page 191</a></li> <li>• <a href="#">no-advertise-peer-as on page 717</a></li> </ul>

## aggregate-label

---

<b>Syntax</b>	aggregate-label { community <i>community-name</i> ; }
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp family inet labeled-unicast], [edit logical-systems <i>logical-system-name</i> protocols bgp family inet6 labeled-unicast], [edit logical-systems <i>logical-system-name</i> protocols bgp family inet-vpn unicast], [edit logical-systems <i>logical-system-name</i> protocols bgp family inet-vpn6 unicast], [edit protocols bgp family inet labeled-unicast], [edit protocols bgp family inet6 labeled-unicast], [edit protocols bgp family inet-vpn unicast], [edit protocols bgp family inet6-vpn unicast]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Specify matching criteria (in the form of a community) such that all routes which match are assigned the same VPN label, selected from one of the several routes in the set defined by this criteria. This reduces the number of VPN labels that the router must consider, and aggregates the received labels.
<b>Options</b>	<b>community <i>community-name</i></b> —Specify the name of the community to which to apply the aggregate label.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring Aggregate Labels for VPNs</i></li> </ul>

## aigp

<b>Syntax</b>	<code>aigp [disable];</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp family inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp family inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet6 labeled-unicast],</p> <p>[edit protocols bgp family inet labeled-unicast],</p> <p>[edit protocols bgp family inet6 labeled-unicast],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> family inet labeled-unicast] ,</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> family inet6 labeled-unicast] ,</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet labeled-unicast],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet6 labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp family inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp family inet6 labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family inet6 labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family inet6 labeled-unicast]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.1.
<b>Description</b>	<p>Enable the accumulated interior gateway protocol (AIGP) BGP attribute on a protocol family. Configuring AIGP on a particular family enables sending and receiving of the AIGP attribute on that family.</p> <p>The AIGP attribute enables deployments in which a single administration can run several contiguous BGP autonomous systems (ASs). Such deployments allow BGP to make routing decisions based on the IGP metric. With AIGP enabled, BGP can select paths based on IGP metrics. This enables BGP to choose the shortest path between two nodes,</p>

even though the nodes might be in different ASs. The AIGP attribute is particularly useful in networks that use tunneling to deliver a packet to its BGP next hop. Such is the case with MPLS label-switched paths.

**Options**    **disable**—Explicitly disables AIGP.

**Default:** Disabled, meaning that the device does not send an AIGP attribute and silently discards a received AIGP attribute.

**Required Privilege**    routing—To view this statement in the configuration.

**Level**    routing-control—To add this statement to the configuration.

- Related Documentation**
- [Example: Configuring the Accumulated IGP Attribute for BGP on page 151](#)
  - [aigp-originate on page 616](#)

## aigp-originate

---

<b>Syntax</b>	<code>aigp-originate <i>distance</i>;</code>
<b>Hierarchy Level</b>	<code>[edit logical-systems <i>logical-system-name</i> policy-options policy-statement <i>policy-name</i> term <i>term-name</i> then],</code> <code>[edit logical-systems <i>logical-system-name</i> policy-options policy-statement <i>policy-name</i> then],</code> <code>[edit policy-options policy-statement <i>policy-name</i> term <i>term-name</i> then],</code> <code>[edit policy-options policy-statement <i>policy-name</i> then]</code>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.1.
<b>Description</b>	<p>Originate an accumulated interior gateway protocol (AIGP) BGP attribute for a given prefix by export policy, using the <b>aigp-originate</b> policy action.</p> <p>To originate an AIGP attribute, you need configure the policy action on only one node. The AIGP attribute is readadvertised if the neighbors are AIGP enabled with the <b>aigp</b> statement in the BGP configuration.</p>
<b>Default</b>	<p>If you omit the <b>aigp-originate</b> policy action, the node still readadvertises the AIGP BGP attribute if AIGP is enabled in the BGP configuration. However, the node does not originate its own AIGP attribute for local prefixes.</p> <p>As the route is readadvertised by downstream nodes, the cost of the AIGP attribute reflects the IGP distance to the prefix (zero + IGP distance or configured distance + IGP distance).</p>
<b>Options</b>	<p><b><i>distance</i></b>—(Optional) Associate an initial cost when advertising a local prefix with the AIGP BGP attribute.</p> <p><b>Range:</b> 0 through 4,294,967,295</p> <p><b>Default:</b> The initial cost associated with the AIGP attribute for a local prefix is zero. The <b><i>distance</i></b> option overrides the default zero value for the initial cost.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring the Accumulated IGP Attribute for BGP on page 151</a></li><li>• <a href="#">aigp on page 614</a></li></ul>

## algorithm (BGP BFD Authentication)

<b>Syntax</b>	<code>algorithm <i>algorithm-name</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit protocols bgp bgp bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bgp bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	Configure the algorithm used to authenticate the specified BFD session.
<b>Options</b>	<p><b><i>algorithm-name</i></b>—Authentication algorithm name: <b>simple-password</b>, <b>keyed-md5</b>, <b>keyed-sha-1</b>, <b>meticulous-keyed-md5</b>, <b>meticulous-keyed-sha-1</b>.</p> <p><b>simple-password</b>—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.</p> <p><b>keyed-md5</b>—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.</p>

**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 can 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 can take additional time to authenticate the session.

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.

- |                              |  |
|------------------------------|--|
| <b>Related Documentation</b> | <ul style="list-style-type: none"><li>• <i>Example: Configuring BFD Authentication for Static Routes</i></li><li>• <a href="#">Example: Configuring BGP Route Authentication on page 449</a></li><li>• <a href="#">Example: Configuring EBGp Multihop Sessions on page 241</a></li><li>• <a href="#">authentication on page 621</a></li><li>• <a href="#">bfd-liveness-detection on page 627</a></li><li>• <a href="#">key-chain on page 676</a></li><li>• <a href="#">loose-check on page 693</a></li></ul> |
|------------------------------|--|

## allow

<b>Syntax</b>	<code>allow (all   [ <i>network/mask-length</i> ] );</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> ], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> ], [edit protocols bgp <b>group</b> <i>group-name</i> ], [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> ]
<b>Release Information</b>	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 11.3 for the QFX Series.
<b>Description</b>	Implicitly configure BGP peers, allowing peer connections from any of the specified networks or hosts. To configure multiple BGP peers, configure one or more networks and hosts within a single <b>allow</b> statement or include multiple <b>allow</b> statements.
<div>  <b>NOTE:</b> You cannot define a BGP group with dynamic peers with BGP authentication enabled. </div>	
<b>Options</b>	<b>all</b> —Allow all addresses, which is equivalent to <b>0.0.0.0/0</b> (or <b>::/0</b> ).  <b>network/mask-length</b> —IPv6 or IPv4 network number of a single address or a range of allowable addresses for BGP peers, followed by the number of significant bits in the subnet mask.
<div>  <b>NOTE:</b> You cannot define a BGP group with dynamic peers with authentication enabled. </div>	
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><a href="#">neighbor on page 713</a></li> </ul>

## as-override

<b>Syntax</b>	as-override;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Compare the AS path of an incoming advertised route with the AS number of the BGP peer under the group and replace all occurrences of the peer AS number in the AS path with its own AS number before advertising the route to the peer.



**NOTE:** The **as-override** statement is specific to a particular BGP group. This statement does not affect peers from the same remote AS configured in different groups.

Enabling the AS override feature allows routes originating from an AS to be accepted by a router residing in the same AS. Without AS override enabled, the routing device refuses the route advertisement once the AS path shows that the route originated from its own AS. This is done by default to prevent route loops. The **as-override** statement overrides this default behavior.

Note that enabling the AS override feature may result in routing loops. Use this feature only for specific applications that require this type of behavior, and in situations with strict network control. One application is the IGP protocol between the provider edge routing device and the customer edge routing device in a virtual private network.

<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring a Layer 3 VPN with Route Reflection and AS Override on page 191</a></li> <li>• <i>Junos OS VPNs Library for Routing Devices</i></li> </ul>

## authentication (BGP BFD Liveness Detection)

<b>Syntax</b>	<pre> authentication {   algorithm <i>algorithm-name</i>;   key-chain <i>key-chain-name</i>;   loose-check ; } </pre>
<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection], [edit protocols bgp bgp bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor   <i>address</i> bfd-liveness-detection] </pre>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	<p>Specify the router and route authentication to mitigate the risk of being attacked by a machine or router that has been configured to share incorrect routing information with another router. Router and route authentication enables routers to share information only if they can verify that they are talking to a trusted source, based on a password (key). In this method, a hashed key is sent along with the route being sent to another router. The receiving router compares the sent key to its own configured key. If they are the same, the receiving router accepts the route.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BFD for Static Routes</a></li> <li>• <a href="#">Example: Configuring BFD Authentication for Static Routes</a></li> <li>• <a href="#">Example: Configuring BGP Route Authentication on page 449</a></li> <li>• <a href="#">algorithm on page 617</a></li> </ul>

- [bfd-liveness-detection on page 627](#)
- [key-chain on page 676](#)
- [loose-check on page 693](#)

## authentication-algorithm

<b>Syntax</b>	<code>authentication-algorithm <i>algorithm</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],          [edit logical-systems <i>logical-system-name</i> protocols ldp session <i>session-address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols ldp session <i>session-address</i>],          [edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>],          [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station</b> <i>station-name</i>],          [edit protocols bgp],          [edit protocols bgp <b>group</b> <i>group-name</i>],          [edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit protocols ldp session <i>session-address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols ldp session <i>session-address</i>],          [edit routing-options <b>bmp</b>],          [edit routing-options bmp <b>station</b> <i>station-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 7.6.          Statement introduced for BGP in Junos OS Release 8.0.          Statement introduced in Junos OS Release 9.0 for EX Series switches.          Statement introduced in Junos OS Release 11.3 for the QFX Series.          Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.          Statement introduced for BMP in Junos OS Release 13.3.</p>
<b>Description</b>	Configure an authentication algorithm type.
<b>Options</b>	<p><b><i>algorithm</i></b>—Specify one of the following types of authentication algorithms:</p> <ul style="list-style-type: none"> <li><b>aes-128-cmac-96</b>—Cipher-based message authentication code (AES128, 96 bits).</li> <li><b>hmac-sha-1-96</b>—Hash-based message authentication code (SHA1, 96 bits).</li> <li><b>md5</b>—Message digest 5.</li> </ul> <p><b>Default:</b> <b>hmac-sha-1-96</b></p>



**NOTE:** The default is not displayed in the output of the `show bgp bmp` command unless a key or key-chain is also configured.

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation**

- [Example: Configuring Route Authentication for BGP on page 450](#)
- [Configuring BGP Monitoring Protocol Version 3 on page 577](#)

## authentication-key (Protocols BGP and BMP)

<b>Syntax</b>	authentication-key <i>key</i> ;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name neighbor address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address</i>],          [edit logical-systems <i>logical-system-name</i> routing-options bmp],          [edit logical-systems <i>logical-system-name</i> routing-options bmp <i>station station-name</i>],          [edit protocols bgp],          [edit protocols bgp <i>group group-name</i>],          [edit protocols bgp <i>group group-name neighbor address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address</i>],          [edit routing-options bmp],          [edit routing-options bmp <i>station station-name</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p>Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.</p> <p>Statement introduced for BMP version 3 in Junos OS Release 13.3.</p>
<b>Description</b>	Configure an MD5 authentication key (password). Neighboring routing devices use the same password to verify the authenticity of BGP packets sent from this system.
<b>Options</b>	<i>key</i> —Authentication password. It can be up to 126 characters. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (" ").
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Route Authentication for BGP on page 450</a></li> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## authentication-key-chain (Protocols BGP and BMP)

<b>Syntax</b>	<code>authentication-key-chain <i>key-chain</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station</b> <i>station-name</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-options <b>bmp</b>],</p> <p>[edit routing-options bmp <b>station</b> <i>station-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.0.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p> <p>Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.</p> <p>Statement introduced for BMP in Junos OS Release 13.3.</p>
<b>Description</b>	Apply and enable an authentication keychain to the routing device. Note that the referenced key chain must be defined. When configuring the authentication key update feature for BGP, you cannot commit the <b>0.0.0.0/allow</b> statement with authentication keys or key chains. The CLI issues a warning and fails to commit the configuration.
<b>Options</b>	<b>key-chain</b> —Authentication keychain name. It can be up to 126 characters. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (" ").
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Route Authentication for BGP on page 450</a></li> <li>• <a href="#">Example: Configuring BFD Authentication for Static Routes</a></li> <li>• <a href="#">Configuring the Authentication Key Update Mechanism for BGP and LDP Routing Protocols</a></li> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## auto-discovery-only

<b>Syntax</b>	auto-discovery-only;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp family l2vpn],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family l2vpn],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family l2vpn],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp family l2vpn],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> family l2vpn],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family l2vpn],</p> <p>[edit protocols bgp family l2vpn],</p> <p>[edit protocols bgp group <i>group-name</i> family l2vpn],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> family l2vpn],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp family l2vpn],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> family l2vpn],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family l2vpn]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 10.4R2.
<b>Description</b>	<p>Enable the router to process only the autodiscovery network layer reachability information (NLRI) update messages for VPWS and LDP-based Layer 2 VPN and VPLS update messages (BGP_L2VPN_AD_NLRI) (FEC 129).</p> <p>Specifically, the <b>auto-discovery-only</b> statement notifies the routing process (rpd) to expect autodiscovery-related NLRI messages so that information can be deciphered and used by LDP, VPLS, and VPWS.</p> <p>The <b>auto-discovery-only</b> statement must be configured on all provider edge (PE) routers in a VPLS or in a VPWS. If you configure route reflection, the <b>auto-discovery-only</b> statement is also required on provider (P) routers that act as the route reflector in supporting FEC 129-related updates.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Configuring BGP Autodiscovery for LDP VPLS</i></li> <li>• <i>Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups</i></li> <li>• <i>Example: Configuring FEC 129 BGP Autodiscovery for VPWS</i></li> </ul>

## bfd-liveness-detection (Protocols BGP)

**Syntax**

```

bfd-liveness-detection {
    authentication {
        algorithm algorithm-name;
        key-chain key-chain-name;
        loose-check;
    }
    detection-time {
        threshold milliseconds;
    }
    hold-down-interval milliseconds;
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    session-mode (automatic | multihop | single-hop);
    transmit-interval {
        minimum-interval milliseconds;
        threshold milliseconds;
    }
    version (1 | automatic);
}

```

**Hierarchy Level**

```

[edit logical-systems logical-system-name protocols bgp],
[edit logical-systems logical-system-name protocols bgp group group-name],
[edit logical-systems logical-system-name protocols bgp group group-name neighbor address],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols
bgp],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols
bgp group group-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols
bgp group group-name neighbor address],
[edit protocols bgp],
[edit protocols bgp group group-name],
[edit protocols bgp group group-name neighbor address],
[edit routing-instances routing-instance-name protocols bgp],
[edit routing-instances routing-instance-name protocols bgp group group-name],
[edit routing-instances routing-instance-name protocols bgp group group-name neighbor
address]

```

**Release Information**

Statement introduced in Junos OS Release 8.1.

Statement introduced in Junos OS Release 9.0 for EX Series switches.

**detection-time threshold** and **transmit-interval threshold** options introduced in Junos OS Release 8.2

Support for logical routers introduced in Junos OS Release 8.3.

Support for IBGP and multihop EBGP sessions introduced in Junos OS Release 8.3.

**holddown-interval** statement introduced in Junos OS Release 8.5. You can configure this statement only for EBGP peers at the **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level.

**no-adaptation** statement introduced in Junos OS Release 9.0.

Support for BFD authentication introduced in Junos OS Release 9.6.

Support for BFD on IPv6 interfaces with BGP introduced in Junos OS Release 11.2.  
Statement introduced in Junos OS Release 12.1 for the QFX Series.

**Description** Configure bidirectional failure detection (BFD) timers and authentication for BGP.

For IBGP and multihop EBGP support, configure the **bfd-liveness-detection** statement at the global **[edit bgp protocols]** hierarchy level. You can also configure IBGP and multihop support for a routing instance or a logical system.

**Options** **authentication algorithm** *algorithm-name* (Optional)—Configure the algorithm used to authenticate the specified BFD session: **simple-password**, **keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**.

**authentication key-chain** *key-chain-name* (Optional)—Associate a security key with the specified BFD session using the name of the security keychain. The keychain name must match one of the keychains configured in the **authentication-key-chains key-chain** statement at the **[edit security]** hierarchy level.

**authentication loose-check**—(Optional) Configure loose authentication checking on the BFD session. Use only for transitional periods when authentication may not be configured at both ends of the BFD session.

**detection-time threshold** *milliseconds* (Optional)—Configure a threshold. When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

**holddown-interval** *milliseconds* (Optional)—Configure an interval specifying how long a BFD session must remain up before a state change notification is sent. When you configure the hold-down interval for the BFD protocol for EBGp, the BFD session is unaware of the BGP session during this time. In this case, if the BGP session goes down during the configured hold-down interval, BFD already assumes it is down and does not send a state change notification. The **holddown-interval** statement is supported only for EBGp peers at the **[edit protocols bgp group group-name neighbor address]** hierarchy level. If the BFD session goes down and then comes back up during the configured hold-down interval, the timer is restarted. You must configure the hold-down interval on both EBGp peers. If you configure the hold-down interval for a multihop EBGp session, you must also configure a local IP address by including the **local-address** statement at the **[edit protocols bgp group group-name]** hierarchy level.

**Range:** 0 through 255,000

**Default:** 0

**minimum-interval** *milliseconds* (Required)—Configure the minimum intervals at which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session. This value represents the minimum interval at which the local routing device transmits hello packets as well as the minimum interval that 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. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately (using the **minimum-receive-interval** and **transmit-interval** statements).

**Range:** 1 through 255,000

**minimum-receive-interval** *milliseconds* (Optional)—Configure only 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.

**Range:** 1 through 255,000

**multiplier *number*** (Optional)—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** (Optional)—Configure BFD sessions not to adapt to changing network conditions. We recommend that you not disable BFD adaptation unless it is preferable to not to have BFD adaptation enabled in your network.

**transmit-interval threshold *milliseconds*** (Optional)—Configure a threshold. When the BFD session 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*** (Optional)—Configure only the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session.

**Range:** 1 through 255,000

**version** (Optional)—Configure the BFD version to detect.

**Range:** 1 or **automatic** (autodetect the BFD version)

**Default:** **automatic**

The remaining statements are explained separately.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
---------------------------------	---

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring BFD for Static Routes</a></li><li>• <a href="#">Example: Configuring BFD Authentication for Static Routes</a></li><li>• <a href="#">Example: Configuring BFD on Internal BGP Peer Sessions on page 350</a></li><li>• <a href="#">Example: Configuring BFD Authentication for BGP on page 360</a></li><li>• <a href="#">Understanding BFD for BGP on page 349</a></li></ul>
------------------------------	---


## bgp

<b>Syntax</b>	<code>bgp { ... }</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit protocols], [edit routing-instances <i>routing-instance-name</i> protocols]
<b>Release Information</b>	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 11.3 for the QFX Series.
<b>Description</b>	Enable BGP on the routing device or for a routing instance.
<b>Default</b>	BGP is disabled.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">BGP Feature Guide for Routing Devices</a></li> </ul>

## bgp-error-tolerance (Protocols BGP)

<b>Syntax</b>	<code>bgp-error-tolerance {     malformed-route-limit <i>number</i>;     malformed-update-log-interval <i>seconds</i>;     no-malformed-route-limit; }</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> ], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> ], [edit protocols bgp], [edit protocols bgp group <i>group-name</i> ], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2.
<b>Description</b>	Enable error handling for BGP update messages.  The remaining statements are explained separately.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Error Handling for BGP Update Messages on page 525</a></li> <li>• <a href="#">Example: Configuring Error Handling for BGP Update Messages on page 527</a></li> </ul>

## bgp-orf-cisco-mode

<b>Syntax</b>	<code>bgp-orf-cisco-mode;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> routing-options <b>outbound-route-filter</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options <b>outbound-route-filter</b>],</p> <p>[edit protocols bgp <b>outbound-route-filter</b>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>outbound-route-filter</b>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>outbound-route-filter</b>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>outbound-route-filter</b>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>outbound-route-filter</b>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>outbound-route-filter</b>],</p> <p>[edit routing-instances <i>routing-instance-name</i> routing-options <b>outbound-route-filter</b>],</p> <p>[edit routing-options <b>outbound-route-filter</b>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.2.</p> <p>Statement introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Support for the BGP group and neighbor hierarchy levels introduced in Junos OS Release 9.2.</p> <p>Support for the BGP group and neighbor hierarchy levels introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	Enable interoperability with routing devices that use the vendor-specific outbound route filter compatibility code of 130 and code type of 128.
	<p> <b>NOTE:</b> To enable interoperability for all BGP peers configured on the routing device, include the statement at the [edit routing-options outbound-route-filter] hierarchy level.</p>
<b>Default</b>	Disabled
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 237](#)

## bmp

```
Syntax  bmp {
    authentication-algorithm (aes-128-cmac-96 | hmac-sha-1-96 | md5);
    authentication-key key;
    authentication-key-chain authentication-key-chain;
    connection-mode (active | passive);
    hold-down {
        seconds;
        flaps flaps;
        period seconds;
    }
    initiation-message text;
    local-address address;
    local-port port;
    monitor (disable | enable);
    priority (high | low | medium);
    route-monitoring {
        none;
        post-policy {
            exclude-non-eligible;
        }
        pre-policy {
            exclude-non-feasible;
        }
    }
}
station station-name {
    authentication-algorithm (aes-128-cmac-96 | hmac-sha-1-96 | md5);
    authentication-key key;
    authentication-key-chain authentication-key-chain;
    connection-mode (active | passive);
    hold-down {
        seconds;
        flaps flaps;
        period seconds;
    }
    initiation-message text;
    local-address address;
    local-port port;
    monitor (disable | enable);
    priority (high | low | medium);
    route-monitoring {
        none;
        post-policy {
            exclude-non-eligible;
        }
        pre-policy {
            exclude-non-feasible;
        }
    }
}
station-address (ip-address | name);
station-port port-number;
statistics-timeout seconds;
traceoptions {
```

```

        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier>;
    }
}
station-address (ip-address | name);
station-port port-number;
statistics-timeout seconds;
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
}
}

```

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols [bgp](#)],  
 [edit logical-systems *logical-system-name* protocols bgp [group](#) *group-name*],  
 [edit logical-systems *logical-system-name* protocols bgp group *group-name* [neighbor](#) *address*],  
 [edit logical-systems *logical-system-name* routing-options],  
 [edit protocols [bgp](#)],  
 [edit protocols bgp [group](#) *group-name*],  
 [edit protocols bgp group *group-name* [neighbor](#) *address*],  
 [edit routing-options]

**Release Information** Statement introduced in Junos OS Release 9.5.  
 Statement introduced in Junos OS Release 9.5 for EX Series switches.  
 Statement introduced in Junos OS Release 12.3 for ACX Series routers.  
 Support for BMP version 3 introduced in Junos OS Release 13.3.

**Description** Configure the BGP Monitoring Protocol (BMP), which enables the routing device to collect data from the BGP Adjacency-RIB-In routing tables and periodically send that data to a monitoring station.

**Options** 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 the BGP Monitoring Protocol on page 580](#)

## cluster

<b>Syntax</b>	<code>cluster <i>cluster-identifier</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Specify the cluster identifier to be used by the route reflector cluster in an internal BGP group.



### CAUTION:

If you configure both route reflection and VPNs on the same routing device, the following modifications to the route reflection configuration cause current BGP sessions to be reset:

- Adding a cluster ID—If a BGP session shares the same AS number with the group where you add the cluster ID, all BGP sessions are reset regardless of whether the BGP sessions are contained in the same group.
- Creating a new route reflector—If you have an IBGP group with an AS number and create a new route reflector group with the same AS number, all BGP sessions in the IBGP group and the new route reflector group are reset.



**NOTE:** If you change the address family specified in the [edit protocols bgp family] hierarchy level, all current BGP sessions on the routing device are dropped and then reestablished.

<b>Options</b>	<i>cluster-identifier</i> —4-byte identifier (such as an IPv4 address).
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BGP Route Reflectors on page 419</a></li> <li>• <a href="#">Understanding External BGP Peering Sessions on page 33</a></li> <li>• <a href="#">no-client-reflect on page 719</a></li> </ul>

## connection-mode

<b>Syntax</b>	connection-mode (active   passive);
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <a href="#">bmp</a> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <a href="#">station station-name</a> ], [edit routing-options <a href="#">bmp</a> ], [edit routing-options bmp <a href="#">station station-name</a> ]
<b>Release Information</b>	Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced for BMP in Junos OS Release 13.3.
<b>Description</b>	Specifies whether the BMP station connection is <b>active</b> or <b>passive</b> .
<b>Options</b>	<p><b>active</b>—BMP initiates the connection to the BMP station.</p> <p><b>passive</b>—BMP does not initiate a connection the BMP station. However, it does listen for a connection request from active BMP stations and will connect if a station is available.</p>
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## damping (Protocols BGP)

<b>Syntax</b>	damping;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family <i>family</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> family <i>family</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family <i>family</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp family <i>family</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> family <i>family</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> family <i>family</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p>Support for flap damping at the address family level introduced in Junos OS Release 12.2.</p>
<b>Description</b>	<p>Enable route flap damping. BGP route flapping describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. Flap damping reduces the number of update messages sent between BGP</p>

peers, thereby reducing the load on these peers, without adversely affecting the route convergence time for stable routes.

You typically apply flap damping to external BGP (EBGP) routes (that is, to routes in different ASs). You can also apply it within a confederation, between confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.) The exception to this rule is when flap damping is applied at the address family level. When you apply flap damping at the address family level, it works for both IBGP and EBGP.

**Default** Flap damping is disabled on the routing device.

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation**

- [Examples: Configuring BGP Flap Damping on page 505](#)
- [Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family on page 515](#)

## description (Protocols BGP)

---

<b>Syntax</b>	<code>description text-description;</code>
<b>Hierarchy Level</b>	<code>[edit logical-systems logical-system-name protocols bgp],</code> <code>[edit logical-systems logical-system-name protocols bgp group group-name],</code> <code>[edit logical-systems logical-system-name protocols bgp group group-name neighbor address],</code> <code>[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp],</code> <code>[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp group group-name],</code> <code>[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp group group-name neighbor address],</code> <code>[edit protocols bgp],</code> <code>[edit protocols bgp group group-name],</code> <code>[edit protocols bgp group group-name neighbor address],</code> <code>[edit routing-instances routing-instance-name protocols bgp],</code> <code>[edit routing-instances routing-instance-name protocols bgp group group-name],</code> <code>[edit routing-instances routing-instance-name protocols bgp group group-name neighbor address]</code>
<b>Release Information</b>	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 11.3 for the QFX Series.
<b>Description</b>	Provide a description of the global, group, or neighbor configuration. If the text includes one or more spaces, enclose it in quotation marks (" "). The text is displayed in the output of the <b>show</b> command and has no effect on the configuration.
<b>Options</b>	<i>text-description</i> —Text description of the configuration. It is limited to 255 characters.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li><i>BGP Feature Guide for Routing Devices</i></li></ul>

## detection-time (BFD Liveness Detection)

<b>Syntax</b>	<pre> detection-time {     threshold milliseconds; } </pre>
<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   l2vpn oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls oam bfd-liveness-detection], [edit protocols bgp bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor   <i>address</i> bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam   bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i>   neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection] </pre>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.2.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPNs and VPLS.</p>
<b>Description</b>	<p>Enable BFD failure detection. The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance</p>

is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

The remaining statement is explained separately.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring BFD for Layer 2 VPN and VPLS</a></li><li>• <a href="#">Example: Configuring BFD for BGP on page 349</a></li><li>• <a href="#">bfd-liveness-detection on page 627</a></li><li>• <a href="#">threshold on page 770</a></li></ul>

---

## disable (Protocols BGP)

---

<b>Syntax</b>	disable;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp]
<b>Release Information</b>	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 11.3 for the QFX Series.
<b>Description</b>	Disable BGP on the system.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

## disable (BGP Graceful Restart)

<b>Syntax</b>	disable;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp graceful-restart], [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> graceful-restart], [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> graceful-restart], [edit protocols bgp graceful-restart], [edit protocols bgp <b>group</b> <i>group-name</i> graceful-restart], [edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> graceful-restart]
<b>Release Information</b>	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.
<b>Description</b>	Disable graceful restart for BGP. Graceful restart allows a routing device undergoing a restart to inform its adjacent neighbors and peers of its condition.



**NOTE:** When you disable graceful restart at one level in the configuration statement hierarchy, it is also disabled at lower levels in the same hierarchy. For example, if you disable graceful restart at the [edit protocols bgp group *group-name*] hierarchy level, it is disabled for all the peers in the group. Therefore, if you want to enable graceful restart for some peers in a group and disable it for others, enable graceful restart at the [edit protocols bgp group *group-name*] hierarchy level and disable graceful restart for each peer at the [edit protocols bgp group *group-name* neighbor *address*] hierarchy level.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring Graceful Restart Options for BGP</i></li> <li>• <i>Configuring Graceful Restart for QFabric Systems</i></li> <li>• <a href="#">graceful-restart on page 654</a></li> </ul>

## explicit-null (Protocols BGP)

<b>Syntax</b>	explicit-null;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols mpls],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols ldap],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> protocols ldap],</p> <p>[edit protocols mpls],</p> <p>[edit protocols bgp <i>family</i> inet labeled-unicast],</p> <p>[edit protocols bgp <i>family</i> inet6 labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> <i>family</i> inet labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet labeled-unicast]</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit protocols ldap],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp <i>family</i> inet labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp <i>family</i> inet6 labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <i>family</i> inet6 labeled-unicast],</p> <p>[edit routing-instances <i>instance-name</i> protocols ldap]</p>
<b>Release Information</b>	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Advertise label 0 to the egress routing device of an LSP.

<b>Default</b>	If you do not include the <b>explicit-null</b> statement in the configuration, label 3 (implicit null) is advertised.
<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Advertising Explicit Null Labels to BGP Peers</i></li></ul>

## export (Protocols BGP)

<b>Syntax</b>	<code>export [ <i>policy-names</i> ];</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Apply one or more policies to routes being exported from the routing table into BGP.</p> <p>If you specify more than one policy, they are evaluated in the order specified, from left to right, and the first matching filter is applied to the route. If no routes match the filters, the routing table exports into BGP only the routes that it learned from BGP. If an action specified in one of the policies manipulates a route characteristic, the policy framework software carries the new route characteristic forward during the evaluation of the remaining policies. For example, if the action specified in the first policy of a chain sets a route's metric to 500, this route matches the criterion of <b>metric 500</b> defined in the next policy.</p>
<b>Options</b>	<i>policy-names</i> —Name of one or more policies.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Route Advertisement on page 233</a></li> <li>• <a href="#">Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</a></li> <li>• <a href="#">import on page 667</a></li> </ul>

## family (Protocols BGP)

```
Syntax  family {
    (inet | inet6 | inet-vpn | inet6-vpn | iso-vpn) {
        (any | flow | labeled-unicast | multicast | unicast) {
            accepted-prefix-limit {
                maximum number;
                teardown <percentage-threshold> idle-timeout (forever | minutes);
            }
            add-path {
                send {
                    path-count number;
                    prefix-policy [ policy-names ];
                }
                receive;
            }
            aigp [disable];
            loops number;
            prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            protection;
            rib-group group-name;
            topology name {
                community {
                    target identifier;
                }
            }
        }
        flow {
            no-validate policy-name;
        }
        labeled-unicast {
            accepted-prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            aggregate-label {
                community community-name;
            }
            explicit-null {
                connected-only;
            }
            prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            resolve-vpn;
            rib (inet.3 | inet6.3);
            rib-group group-name;
            traffic-statistics {
                file filename <world-readable | no-world-readable>;
                interval seconds;
            }
        }
    }
}
```

```
    }
  }
  route-target {
    accepted-prefix-limit {
      maximum number;
      proxy-generate <route-target-policy route-target-policy-name>;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    advertise-default;
    external-paths number;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
  }
  (evpn | inet-mdt | inet-mvpn | inet6-mvpn | l2vpn) {
    signaling {
      accepted-prefix-limit {
        maximum number;
        teardown <percentage-threshold> idle-timeout (forever | minutes);
      }
      add-path {
        send {
          path-count number;
          prefix-policy [ policy-names ];
        }
        receive;
      }
      aigp [disable];
      damping;
      loops number;
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      rib-group group-name;
    }
  }
}
```

<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>   <b>neighbor</b> <i>address</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp <b>group</b> <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>], [edit protocols bgp], [edit protocols bgp <b>group</b> <i>group-name</i>], [edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>   <b>neighbor</b> <i>address</i>] </pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>inet-mvpn</b> and <b>inet6-mvpn</b> statements introduced in Junos OS Release 8.4.</p> <p><b>inet-mdt</b> statement introduced in Junos OS Release 9.4.</p> <p>Support for the <b>loops</b> statement introduced in Junos OS Release 9.6.</p> <p><b>evpn</b> statement introduced in Junos OS Release 13.2.</p>
<b>Description</b>	<p>Enable multiprotocol BGP (MP-BGP) by configuring BGP to carry network layer reachability information (NLRI) for address families other than unicast IPv4, to specify MP-BGP to carry NLRI for the IPv6 address family, or to carry NLRI for VPNs.</p>

- Options**
- any**—Configure the family type to be both unicast and multicast.
  - evpn**—Configure NLRI parameters for Ethernet VPNs (EVPNs).
  - inet**—Configure NLRI parameters for IPv4.
  - inet6**—Configure NLRI parameters for IPv6.
  - inet-mdt**—Configure NLRI parameters for the multicast distribution tree (MDT) subaddress family identifier (SAFI) for IPv4 traffic in Layer 3 VPNs.
  - inet-mvpn**—Configure NLRI parameters for IPv4 for multicast VPNs.
  - inet6-mvpn**—Configure NLRI parameters for IPv6 for multicast VPNs.
  - inet-vpn**—Configure NLRI parameters for IPv4 for Layer 3 VPNs.
  - inet6-vpn**—Configure NLRI parameters for IPv6 for Layer 3 VPNs.
  - iso-vpn**—Configure NLRI parameters for IS-IS for Layer 3 VPNs.
  - l2vpn**—Configure NLRI parameters for IPv4 for MPLS-based Layer 2 VPNs and VPLS.
  - labeled-unicast**—Configure the family type to be labeled-unicast. This means that the BGP peers are being used only to carry the unicast routes that are being used by labeled-unicast for resolving the labeled-unicast routes. This statement is supported only with **inet** and **inet6**.
  - multicast**—Configure the family type to be multicast. This means that the BGP peers are being used only to carry the unicast routes that are being used by multicast for resolving the multicast routes.
  - unicast**—Configure the family type to be unicast. This means that the BGP peers only carry the unicast routes that are being used for unicast forwarding purposes. The default family type is **unicast**.

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**
- *Configuring IBGP Sessions Between PE Routers in VPNs*
  - [Understanding Multiprotocol BGP on page 537](#)
  - *autonomous-system*
  - [local-as on page 684](#)

## file (Tracing for Origin AS Validation)

<b>Syntax</b>	file <i>filename</i> <files <i>number</i> > <size <i>size</i> > <world-readable   no-world-readable>;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation traceoptions], [edit logical-systems <i>logical-system-name</i> routing-options validation traceoptions], [edit routing-instances <i>instance-name</i> routing-options validation traceoptions], [edit routing-options validation traceoptions]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure the file settings for tracing resource public key infrastructure (RPKI) BGP route validation.
<b>Options</b>	<p><b><i>filename</i></b> —Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory <code>/var/log</code>.</p> <p><b><i>files number</i></b>—(Optional) Maximum number of trace files. When a trace file named <b><i>trace-file</i></b> reaches its maximum size, it is renamed <b><i>trace-file.0</i></b>, then <b><i>trace-file.1</i></b>, and so on, until the maximum number of trace files is reached (<b><i>xk</i></b> to specify KB, <b><i>xm</i></b> to specify MB, or <b><i>xg</i></b> to specify gigabytes), at which point the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the <b><i>size</i></b> option.</p> <p><b>Range:</b> 2 through 1000 files</p> <p><b>Default:</b> 3 files</p> <p><b><i>no-world-readable</i></b>—(Optional) Restrict file access to the user who created the file.</p> <p><b><i>size size</i></b>—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named <b><i>trace-file</i></b> reaches its maximum size, it is renamed <b><i>trace-file.0</i></b>, then <b><i>trace-file.1</i></b>, 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 also must specify a maximum file size with the <b><i>files</i></b> option.</p> <p><b>Syntax:</b> <b><i>xk</i></b> to specify KB, <b><i>xm</i></b> to specify MB, or <b><i>xg</i></b> to specify GB</p> <p><b>Range:</b> 10 KB through 1 GB</p> <p><b>Default:</b> 128 KB</p> <p><b><i>world-readable</i></b>—(Optional) Enable unrestricted file access.</p>
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## flag (Tracing for Origin AS Validation)

---

<b>Syntax</b>	<code>flag <i>flag</i>;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation traceoptions], [edit logical-systems <i>logical-system-name</i> routing-options validation traceoptions], [edit routing-instances <i>instance-name</i> routing-options validation traceoptions], [edit routing-options validation traceoptions]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure the flags for tracing resource public key infrastructure (RPKI) BGP route validation.
<b>Options</b>	<p><i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple <b>flag</b> statements.</p> <p><b>RPKI BGP Route Validation Tracing Flags</b></p> <ul style="list-style-type: none"><li>• <b>all</b>—Trace all events.</li><li>• <b>error</b>—Trace errored packets.</li><li>• <b>keepalive</b>—RPKI-to-router protocol keepalive messages. If you enable the BGP <b>update</b> flag only, received keepalive messages do not generate a trace message.</li><li>• <b>nsr-synchronization</b>—Nonstop routing synchronization events.</li><li>• <b>packets</b>—All incoming and outgoing packets.</li><li>• <b>state</b>—State transitions.</li><li>• <b>task</b>—Routing protocol task processing.</li><li>• <b>timer</b>—Routing protocol timer processing.</li><li>• <b>update</b>—Update packets. These packets provide routing updates to BGP systems. If you enable only this flag, received keepalive messages do not generate a trace message. Use the <b>keepalive</b> flag to generate a trace message for keepalive messages.</li></ul>
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## flow

<b>Syntax</b>	flow { no-validate <i>policy-name</i> ; }
<b>Hierarchy Level</b>	[edit protocols bgp group <i>group-name</i> family (inet   inet-vpn)], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet-vpn)], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> family (inet   inet-vpn)], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet-vpn)]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Enables BGP to support flow routes.




**NOTE:** This statement is supported for the default instance, VRF instance, and virtual-router instance only. It is configured with the instance-type statement at the [edit routing-instance *instance-name*] hierarchy level. For VPNs, this statement is supported for the default instance only.

The remaining statements are explained separately.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Enabling BGP to Carry Flow-Specification Routes on page 554</a></li> </ul>

## graceful-restart (Protocols BGP)

<b>Syntax</b>	<pre> graceful-restart {   disable;   restart-time seconds;   stale-routes-time seconds; } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],  [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],  [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],  [edit protocols bgp],  [edit protocols bgp <b>group</b> <i>group-name</i>],  [edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<p>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.</p>
<b>Description</b>	<p>Configure graceful restart for BGP. Graceful restart allows a routing device undergoing a restart to inform its adjacent neighbors and peers of its condition. Graceful restart is disabled by default.</p> <p>To configure the duration of the BGP graceful restart period, include the <b>restart-time</b> statement at the [edit protocols bgp graceful-restart] hierarchy level. To set the length of time the router waits to receive messages from restarting neighbors before declaring them down, include the <b>stale-routes-time</b> statement at the [edit protocols bgp graceful-restart] hierarchy level.</p> <hr/> <div>  <p><b>NOTE:</b> If you configure graceful restart after a BGP session has been established, the BGP session restarts and the peers negotiate graceful restart capabilities.</p> </div> <hr/> <p>Configure graceful restart globally at the [edit routing-options] or [edit routing-instances <i>instance-name</i> routing-options] hierarchy level to enable the feature. You cannot enable graceful restart for specific protocols unless graceful restart is also enabled globally. You can, optionally, modify the global settings at the individual protocol level.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.  routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>Configuring Graceful Restart Options for BGP</li> <li>Configuring Graceful Restart for QFabric Systems</li> <li>Junos OS High Availability Library for Routing Devices</li> </ul>

## group (Protocols BGP)

```
Syntax  group group-name {
    advertise-inactive;
    allow [ network/mask-length ];
    authentication-key key;
    cluster cluster-identifier;
    damping;
    description text-description;
    export [ policy-names ];
    family {
        (inet | inet6 | inet-vpn | inet6-vpn | l2-vpn) {
            (any | multicast | unicast | signaling) {
                accepted-prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                add-path {
                    send {
                        path-count number;
                        prefix-policy [ policy-names ];
                    }
                    receive;
                }
                aigp [disable];
                damping;
                prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                rib-group group-name;
                topology name {
                    community {
                        target identifier;
                    }
                }
            }
        }
        flow {
            no-validate policy-name;
        }
        labeled-unicast {
            accepted-prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            explicit-null {
                connected-only;
            }
            prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            resolve-vpn;
            rib inet.3;
        }
    }
}
```

```

        rib-group group-name;
    }
}
route-target {
    accepted-prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    advertise-default;
    external-paths number;
    prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
    }
}
}
hold-time seconds;
import [ policy-names ];
ipsec-sa ipsec-sa;
keep (all | none);
local-address address;
local-as autonomous-system <private>;
local-preference local-preference;
log-updown;
metric-out metric;
multihop <ttl-value>;
multipath {
    multiple-as;
}
no-aggregator-id;
no-client-reflect;
out-delay seconds;
passive;
peer-as autonomous-system;
preference preference;
remove-private;
tcp-aggressive-transmission;
tcp-mss segment-size;
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
type type;
neighbor address {
    ... peer-specific-options ...
}
}

```

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols bgp],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols  
bgp],  
[edit protocols bgp],  
[edit routing-instances *routing-instance-name* protocols bgp]

<b>Release Information</b>	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 11.3 for the QFX Series.
<b>Description</b>	<p>Define a BGP peer group. BGP peer groups share a common type, peer autonomous system (AS) number, and cluster ID, if present. To configure multiple BGP groups, include multiple <b>group</b> statements.</p> <p>By default, the group's options are identical to the global BGP options. To override the global options, include group-specific options within the <b>group</b> statement.</p> <p>The <b>group</b> statement is one of the statements you must include in the configuration to run BGP on the routing device.</p> <p>Each group must contain at least one peer.</p>
<b>Options</b>	<p><b>group-name</b>—Name of the BGP group.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>BGP Feature Guide for Routing Devices</i></li></ul>

## group (Origin Validation for BGP)

---

Syntax	<pre>group group-name {     max-sessions number;     session address {         hold-time seconds;         local-address local-ip-address;         port port-number;         preference number;         record-lifetime seconds;         refresh-time seconds;     } }</pre>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation], [edit logical-systems <i>logical-system-name</i> routing-options validation], [edit routing-instances <i>instance-name</i> routing-options validation], [edit routing-options validation]
Release Information	Statement introduced in Junos OS Release 12.2.
Description	<p>Configure the number of concurrent sessions for each group.</p> <p>Caches are organized in groups. The Junos OS implementation supports up to 63 sessions per group and both IPv4 and IPv6 address families.</p> <p>If the number of sessions in a group exceeds the <b>max-sessions</b> value, the connections are established in order by <b>preference</b> value. A numerically higher preference results in a higher probability for session establishment. The order of session establishment is random among sessions with equal preferences.</p>
Options	<p><b>group-name</b>—Name of the validation group.</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"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## hold-down

<b>Syntax</b>	<pre>hold-down {     seconds;     flaps <i>number</i>;     period <i>seconds</i>; }</pre>
<b>Hierarchy Level</b>	<pre>[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>], [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station</b> <i>station-name</i>], [edit routing-options <b>bmp</b>], [edit routing-options bmp <b>station</b> <i>station-name</i>]</pre>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.3.</p>
<b>Description</b>	<p>If the connection to a BMP station flaps and the <b>hold-down</b> statement is configured, the station is prevented from reconnecting to the device for the specified period of time. A flap is when the TCP session unexpectedly switches from established to non-established. If you alter the configuration of the <b>hold-down</b> statement, the hold down timer and flap counter are reset.</p> <p>You can effectively disable the <b>hold-down</b> statement by setting the <b>flaps</b> option to 10 and the <b>period</b> option to 30 seconds.</p>
<b>Options</b>	<p><b>seconds</b>—Specify the time in seconds to wait before allowing the BMP station to reconnect to the device.</p> <p><b>Default:</b> 600 seconds</p> <p><b>Range:</b> 30 through 65,535 seconds</p> <p><b>flaps <i>number</i></b>—Specify the number of BMP station flaps allowed before terminating the connection to the BMP station and triggering the hold down timer.</p> <p><b>Default:</b> 3 flaps</p> <p><b>Range:</b> 2 to 10 flaps</p> <p><b>period <i>seconds</i></b>—Specify the time in seconds for the BGP station flaps (specified using the <b>flaps</b> option) to occur before triggering the hold down timer. Every time a flap occurs, the number of flaps in the last time period is checked to see if the criteria is met.</p> <p>For example, if you defined the <b>period</b> as 60 seconds and the <b>flaps</b> as 4 and the BGP station flaps just 2 times in a 60 second period, the hold down timer would not be triggered. However, if the BGP station flaps 4 times in a 60 second period, the hold down timer would be triggered.</p> <p><b>Default:</b> 300 seconds</p> <p><b>Range:</b> 30 through 65,535 seconds</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Configuring BGP Monitoring Protocol Version 3 on page 577](#)

## hold-down-interval (BGP BFD Liveness Detection)

<b>Syntax</b>	<code>holddown-interval milliseconds;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],  [edit protocols bgp bfd-liveness-detection],  [edit protocols bgp group <i>group-name</i> bfd-liveness-detection],  [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],  [edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.5.  Statement introduced in Junos OS Release 9.0 for EX Series switches.  Support for BFD authentication introduced in Junos OS Release 9.6.  Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	<p>Configure an interval specifying how long a BFD session must remain up before a state change notification is sent.</p> <p>When you configure the hold-down interval for the BFD protocol for EBGp, the BFD session is unaware of the BGP session during this time. In this case, if the BGP session goes down during the configured hold-down interval, BFD already assumes the BGP session is down and does not send a state change notification. The <b>holddown-interval</b> statement is supported only for EBGp peers at the [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>] hierarchy level. If the BFD session goes down and then comes back up during the configured hold-down interval, the timer is restarted. You must configure the hold-down interval on both EBGp peers. If you configure the hold-down interval for a multihop EBGp session, you must also configure a local IP address by including the <b>local-address</b> statement at the [edit protocols bgp group <i>group-name</i>] hierarchy level.</p>
<b>Options</b>	<p><b>milliseconds</b>—Specify the hold-down interval value.  <b>Range:</b> 0 through 255,000  <b>Default:</b> 0</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.  routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- *Example: Configuring BFD for Static Routes*
  - [bfd-liveness-detection on page 627](#)

## hold-time (Protocols BGP)

<b>Syntax</b>	<code>hold-time seconds;</code>
<b>Hierarchy Level</b>	<pre>[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i> <i>neighbor address</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address</i>], [edit protocols bgp], [edit protocols bgp <i>group group-name</i>], [edit protocols bgp <i>group group-name neighbor address</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i> <i>neighbor address</i>]</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify the hold-time value to use when negotiating a connection with the peer. The hold-time value is advertised in open packets and indicates to the peer the length of time that it should consider the sender valid. If the peer does not receive a keepalive, update, or notification message within the specified hold time, the BGP connection to the peer is closed and routing devices through that peer become unavailable.</p> <p>The hold time is three times the interval at which keepalive messages are sent.</p> <p>BGP on the local routing device uses the smaller of either the local hold-time value or the peer's hold-time value received in the open message as the hold time for the BGP connection between the two peers.</p> <p>Starting in Junos OS Release 12.3, the BGP hold-time value can be zero (0). This implies that the speaker does not expect keepalive messages from its peer to maintain the BGP session. When negotiating between two peers, if one side requests a nonzero hold time and the other requests a zero hold time, the negotiation settles on the nonzero value and keepalive intervals are determined accordingly. Both sides must be set to zero for keepalive messages to stop being sent.</p>
<b>Options</b>	<p><b>seconds</b>—Hold time.</p> <p><b>Range:</b> 10 through 65,535 seconds (or 0 for infinite hold time)</p> <p><b>Default:</b> 90 seconds</p>



**TIP:** When you set a hold-time value of 1 through 19 seconds, we recommend that you also configure the BGP `precision-timers` statement. The `precision-timers` statement ensures that if scheduler slip messages occur, the routing device continues to send keepalive messages. When the `precision-timers` statement is included, keepalive message generation is performed in a dedicated kernel thread, which helps to prevent BGP session flaps.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">BGP Messages Overview on page 7</a></li><li>• <a href="#">precision-timers on page 735</a></li></ul>

## hold-time (Origin Validation for BGP)

<b>Syntax</b>	<code>hold-time seconds;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>][edit routing-options validation group <i>group-name</i> session <i>server-ip-address</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	<p>Specify the length of time in seconds that the session between the routing device and the cache server is to be considered operational without any activity. After the hold time expires, the session is dropped.</p> <p>Reception of any protocol data unit (PDU) from the cache server resets the hold timer. The hold time must be configured to be at least 2 x the <a href="#">refresh-time</a>. If the hold time expires, the session is considered to be down. This, in turn, triggers a session restart event. During a session restart, the routing device attempts to start a session with the cache server that has the numerically highest <a href="#">preference</a>.</p>
<b>Options</b>	<p><b>seconds</b>—Time after which the session is declared down.</p> <p><b>Range:</b> 10 through 3600</p> <p><b>Default:</b> 600</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## idle-after-switch-over

---

<b>Syntax</b>	<code>idle-after-switch-over (forever   seconds);</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> ], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> ], [edit protocols bgp], [edit protocols bgp group <i>group-name</i> ], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.5. Statement introduced in Junos OS Release 9.5 for EX Series switches. Statement introduced in Junos OS Release 11.3 for the QFX Series.
<b>Description</b>	Configure the routing device so that it does not automatically reestablish BGP peer sessions after a nonstop active routing (NSR) switchover. This feature is particularly useful if you are using dynamic routing policies because the dynamic database is not synchronized with the backup Routing Engine when NSR is enabled.
<b>Options</b>	<b>forever</b> —Do not reestablish a BGP peer session after an non-stop routing switchover until the <b>clear bgp neighbor</b> command is issued.  <b>seconds</b> —Do not reestablish a BGP peer session after an non-stop routing switchover until after the specified period. <b>Range:</b> 1 through 4,294,967,295 ( $2^{32} - 1$ )
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Preventing Automatic Reestablishment of BGP Peer Sessions After NSR Switchovers</i></li><li>• <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i></li><li>• <i>Junos OS High Availability Library for Routing Devices</i></li></ul>

## import (Protocols BGP)

<b>Syntax</b>	<code>import [ <i>policy-names</i> ];</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Apply one or more routing policies to routes being imported into the Junos OS routing table from BGP.</p> <p>If you specify more than one policy, they are evaluated in the order specified, from left to right, and the first matching filter is applied to the route. If no match is found, BGP places into the routing table only those routes that were learned from BGP routing devices. The policy framework software evaluates the routing policies in a chain sequentially. If an action specified in one of the policies manipulates a route characteristic, the policy framework software carries the new route characteristic forward during the evaluation of the remaining policies. For example, if the action specified in the first policy of a chain sets a route's metric to 500, this route matches the criterion of <b>metric 500</b> defined in the next policy.</p> <p>It is also important to understand that in Junos OS, although an import policy (inbound route filter) might reject a route, not use it for traffic forwarding, and not include it in an advertisement to other peers, the router retains these routes as hidden routes. These hidden routes are not available for policy or routing purposes. However, they do occupy memory space on the router. A service provider filtering routes to control the amount of information being kept in memory and processed by a router might want the router to entirely drop the routes being rejected by the import policy.</p> <p>Hidden routes can be viewed by using the <b>show route receive-protocol bgp neighbor-address hidden</b> command. The hidden routes can then be retained or dropped from the routing table by configuring the <b>keep all   none</b> statement at the [edit protocols bgp] or [edit protocols bgp group <i>group-name</i>] hierarchy level.</p>

The rules of BGP route retention are as follows:

- By default, all routes learned from BGP are retained, except those where the AS path is looped. (The AS path includes the local AS.)
- By configuring the **keep all** statement, all routes learned from BGP are retained, even those with the local AS in the AS path.
- By configuring the **keep none** statement, all routes received are discarded. When this statement is configured and the inbound policy changes, Junos OS re-advertises all the routes advertised by the peer.

**Options**    *policy-names*—Name of one or more policies.

**Required Privilege**    routing—To view this statement in the configuration.  
**Level**    routing-control—To add this statement to the configuration.

**Related Documentation**

- [Example: Configuring BGP Interactions with IGP on page 220](#)
- [Understanding Route Advertisement on page 233](#)
- *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices*
- [export on page 646](#)

## include-mp-next-hop

<b>Syntax</b>	include-mp-next-hop;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Enable multiprotocol updates to contain next-hop reachability information.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Examples: Configuring Multiprotocol BGP on page 537</a></li> </ul>

## inet-mdt (Signaling)

<b>Syntax</b>	<pre> signaling {   accepted-prefix-limit {     maximum <i>number</i>;     teardown &lt;percentage-threshold&gt; idle-timeout (forever   <i>minutes</i>);   }   add-path {     send {       path-count <i>number</i>;       prefix-policy [ <i>policy-names</i> ];     }     receive;   }   aigp [disable];   loops <i>number</i>;   prefix-limit {     maximum <i>number</i>;     teardown &lt;percentage&gt; &lt;idle-timeout (forever   <i>minutes</i>)&gt;;   }   rib-group <i>group-name</i>; } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems logical-system-name protocols bgp family],  [edit logical-systems logical-system-name protocols bgp group group-name family],  [edit logical-systems logical-system-name protocols bgp group group-name neighbor address family],  [edit protocols bgp family],  [edit protocols bgp group group-name family],  [edit protocols bgp group group-name neighbor address family],  [edit routing-instances instance-name protocols bgp family],  [edit routing-instances instance-name protocols bgp group group-name family],  [edit routing-instances instance-name protocols bgp group group-name neighbor address family]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 9.4.
<b>Description</b>	For draft-rosen 7, on the provider edge router enable BGP intra-AS auto-discovery using MDT-SAFI.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.  routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs</i></li> </ul>

## initiation-message

<b>Syntax</b>	initiation-message <i>text</i> ;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station station-name</b> ], [edit routing-options <b>bmp</b> ], [edit routing-options bmp <b>station station-name</b> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	<p>(Optional) Allows you to specify an initiation message for a type 0 TLV to be sent to the BMP station. The message is transmitted when a BMP station establishes a connection to the device. You can provide some information to the BMP station system administrator (for example, a contact phone number). The initiation message includes a type 1 TLV containing the SNMP sysDescr value specified in RFC 1213 <i>Management Information Base for Network Management of TCP/IP-based internets: MIB-II</i> and a type 2 TLV containing the SNMP sysName value also from RFC 1213. The string in the initiation-message message is UTF-8.</p> <p>The normal time for sending an initiation message is when the BMP session is first established. However, an initiation message change also triggers the transmission of an initiation message to current BMP sessions.</p> <p>Another event that triggers the transmission of an initiation message is when you change in the sysName or sysDescr values in the SNMP configuration. The initiation message is sent to current BMP sessions.</p>
<b>Options</b>	<p><b>text</b>—Specify a character string for a type 0 TLV to send with the initiation message.</p> <p><b>Range:</b> 1 through 255 characters</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## ipsec-sa (Protocols BGP)

---

<b>Syntax</b>	<code>ipsec-sa ipsec-sa;</code>
<b>Hierarchy Level</b>	<code>[edit logical-systems <i>logical-system-name</i> protocols bgp],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i></code> <code>    <b>neighbor</b> <i>address</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code>    bgp],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code>    bgp <b>group</b> <i>group-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code>    bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</code> <code>[edit protocols bgp],</code> <code>[edit protocols bgp <b>group</b> <i>group-name</i>],</code> <code>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i></code> <code>    <b>neighbor</b> <i>address</i>]</code>
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	Specify a security association to BGP peers. You can apply the security association globally for all BGP peers, to a group of peers, or to an individual peer.
<b>Options</b>	<i>ipsec-sa</i> —Security association name.
<b>Required Privilege Level</b>	<code>routing</code> —To view this statement in the configuration. <code>routing-control</code> —To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Using IPsec to Protect BGP Traffic on page 456</a></li></ul>

## iso-vpn

<b>Syntax</b>	<pre>iso-vpn {     unicast {         prefix-limit <i>number</i>;         rib-group <i>group-name</i>;     } }</pre>
<b>Hierarchy Level</b>	<p>[edit protocols bgp <i>family</i>],          [edit protocols bgp group <i>group-name</i> <i>family</i>],          [edit protocols bgp group <i>group-name</i> neighbor <i>address-family</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>family</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <i>family</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address-family</i>]</p>
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	<p>Enable BGP to carry ISO VPN NLRI messages between PE routes connecting a VPN.</p> <p>The remaining statements are explained separately in this chapter.</p>
<b>Default</b>	Disabled.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BGP and CLNS on page 569</a></li> </ul>

## keep

<b>Syntax</b>	keep (all   none);
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Control whether or not Junos OS keeps in memory and hides certain routes.</p> <p>If the <b>keep none</b> statement is used, Junos OS does not retain in memory and hide routes that are rejected because of a BGP import policy. Nor does BGP keep in memory and hide routes that are declared unfeasible due to BGP sanity checks. The <b>keep none</b> statement causes Junos OS to discard from memory the routes that are rejected due to BGP-specific logic or BGP evaluation. When a route is rejected because of some non-BGP-specific reason, the <b>keep none</b> statement has no effect on this route. This rejected route is retained in memory and hidden even though <b>keep none</b> is configured. An example of this type of hidden route is a route for which the protocol nexthop is unresolved.</p> <p>The routing table can retain the route information learned from BGP in one of the following ways:</p> <ul style="list-style-type: none"> <li>• Default (omit the <b>keep</b> statement)—Keep all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.</li> <li>• <b>keep all</b>—Keep all route information that was learned from BGP.</li> <li>• <b>keep none</b>—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking, such as AS path or next hop. When you configure <b>keep none</b> for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.</li> </ul>

In an AS path healing situation, routes with looped paths theoretically could become usable during a soft reconfiguration when the AS path loop limit is changed. However, there is a significant memory usage difference between the default and **keep all**.

Consider the following scenarios:

- A peer readadvertises routes back to the peer from which it learned them.

This can happen in the following cases:

- Another vendor's routing device advertises the routes back to the sending peer.
- The Junos OS peer's default behavior of not readvertising routes back to the sending peer is overridden by configuring **advertise-peer-as**.
- A provider edge (PE) routing device discards any VPN route that does not have any of the expected route targets.

When **keep all** is configured, the behavior of discarding routes received in the above scenarios is overridden.



**CAUTION:** If you add or remove **keep all** or **keep none** and the peer does not support session restart, the associated BGP sessions are restarted (flapped). To determine if a peer supports refresh, check for **Peer supports Refresh capability** in the output of the **show bgp neighbor** command.

<b>Default</b>	By default, BGP retains incoming rejected routes in memory and hides them. If you do not include the <b>keep</b> statement, most routes are retained in the routing table. BGP keeps all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.
<b>Options</b>	<p><b>all</b>—Retain all routes.</p> <p><b>none</b>—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking. When <b>keep none</b> is configured for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">out-delay on page 725</a></li> <li>• <i>Interprovider VPN Example—MP-EBGP Between ISP Peer Routers</i></li> <li>• <a href="#">Example: Configuring Conditional Installation of Prefixes in a Routing Table on page 282</a></li> </ul>

## key-chain (BGP BFD Authentication)

<b>Syntax</b>	<code>key-chain key-chain-name;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit protocols bgp bgp bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> bgp bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	Associate a security key with the specified BFD session using the name of the security keychain. Each key has a unique start time within the keychain. Keychain authentication allows you to change the password information periodically without bringing down peering sessions. This keychain authentication method is referred to as <i>hitless</i> because the keys roll over from one to the next without resetting any peering sessions or interrupting the routing protocol.
<b>Options</b>	<b>key-chain-name</b> —Name of the authentication keychain. The keychain name must match one of the keychains configured with the <b>key-chain key-chain-name</b> statement at the [edit security authentication-key-chain] hierarchy level.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Configuring BFD for Static Routes</i></li> <li>• <i>Example: Configuring BFD Authentication for Static Routes</i></li> <li>• <a href="#">Example: Configuring BFD on Internal BGP Peer Sessions on page 350</a></li> <li>• <a href="#">Example: Configuring BGP Route Authentication on page 449</a></li> </ul>

- [Example: Configuring EBGp Multihop Sessions on page 241](#)

## labeled-unicast (Protocols BGP)

<b>Syntax</b>	<pre> labeled-unicast {     accepted-prefix-limit {         maximum <i>number</i>;         teardown &lt;<i>percentage</i>&gt; &lt;idle-timeout (forever   <i>minutes</i>)&gt;;     }     aggregate-label {         community <i>community-name</i>;     }     explicit-null {         connected-only;     }     prefix-limit {         maximum <i>number</i>;         teardown &lt;<i>percentage</i>&gt; &lt;idle-timeout (forever   <i>minutes</i>)&gt;;     }     protection;     resolve-vpn;     rib (inet.3   inet6.3);     rib-group <i>group-name</i>; } </pre>
<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp <b>family</b> (inet   inet6)], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6)], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6)], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6)], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6)], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6)], [edit protocols bgp <b>family</b> (inet   inet6)], [edit protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6)], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6)], [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6)], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6)], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6)] </pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Configure the family type to be labeled-unicast.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Examples: Configuring Multiprotocol BGP on page 537](#)

## local-address (Protocols BGP)

<b>Syntax</b>	<code>local-address address;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify the address of the local end of a BGP session. This address is used to accept incoming connections to the peer and to establish connections to the remote peer. When none of the operational interfaces are configured with the specified local address, a session with a BGP peer is placed in the idle state.</p> <p>You generally configure a local address to explicitly configure the system's IP address from BGP's point of view. This IP address can be either an IPv6 or IPv4 address. Typically, an IP address is assigned to a loopback interface, and that IP address is configured here.</p> <p>For internal BGP (IBGP) peering sessions, generally the loopback interface (lo0) is used to establish connections between the IBGP peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the IBGP peering session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peering session also goes up and down. Thus, the loopback interface provides fault tolerance in case the physical interface or the link goes down, if the device has link redundancy.</p> <p>When a device peers with a remote device's loopback interface address, the local device expects BGP update messages to come from (be sourced by) the remote device's loopback interface address. The <b>local-address</b> statement enables you to specify the source information in BGP update messages. If you omit the <b>local-address</b> statement, the expected source of BGP update messages is based on the device's source address selection rules, which normally result in the egress interface address being the expected source of update messages. When this happens, the peering session is not established because a mismatch exists between the expected source address (the egress interface</p>

of the peer) and the actual source (the loopback interface of the peer). To ensure that the expected source address matches the actual source address, specify the loopback interface address in the **local-address** statement.



**NOTE:** Although a BGP session can be established when only one of the paired routing devices has **local-address** configured, we strongly recommend that you configure **local-address** on both paired routing devices for IBGP and multihop EBGP sessions. The **local-address** statement ensures that deterministic fixed addresses are used for the BGP session end-points.

If you include the **default-address-selection** statement in the configuration, the software chooses the system default address as the source for most locally generated IP packets. For protocols in which the local address is unconstrained by the protocol specification, for example IBGP and multihop EBGP, if you do not configure a specific local address when configuring the protocol, the local address is chosen using the same methods as other locally generated IP packets.

**Default** If you do not configure a local address, BGP uses the routing device's source address selection rules to set the local address.

**Options** **address**—IPv6 or IPv4 address of the local end of the connection.

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**

- [Example: Configuring Internal BGP Peering Sessions on Logical Systems on page 68](#)
- [Example: Configuring Internal BGP Peer Sessions on page 57](#)
- [Understanding Internal BGP Peering Sessions on page 56](#)
- *router-id*

## local-address (Protocols BMP)

---

<b>Syntax</b>	<code>local-address address;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <a href="#">bmp</a> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <a href="#">station station-name</a> ], [edit routing-options <a href="#">bmp</a> ], [edit routing-options bmp <a href="#">station station-name</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	<p>(Optional) Specifies the IPv4 or IPv6 address for the BMP connection on the device. We recommend that you configure a local address. For both active and passive modes, configure a loopback local address. This provides a consistent local endpoint, is useful for debugging, and assures greater reliability for the BMP connection since it is not tied to a single router interface.</p> <p>For passive mode, specifying a local address is required. It also provides some security against a malicious BMP connection. For active mode, we also recommend configuring a local address to help ensure reliability.</p> <p>If you change the local address, the BMP station connection flaps when you commit the configuration.</p>
<b>Options</b>	<b>address</b> —Specify the IPv4 or IPv6 address for the BMP connection on the local device.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li></ul>

## local-address (Origin Validation for BGP)

<b>Syntax</b>	<code>local-address <i>local-ip-address</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-options validation group <i>group-name</i> session <i>server-ip-address</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure a local IP address of the session. If the local cache server has inbound firewall filtering, it might be necessary to specify a local IP address to use for this session.
<b>Options</b>	<i>local-ip-address</i> —Local IP address to be used for the outgoing connection to the cache server.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## local-as

<b>Syntax</b>	<code>local-as <i>autonomous-system</i> &lt;loops number&gt; &lt;private   alias&gt; &lt;no-prepend-global-as&gt;;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>alias</b> option introduced in Junos OS Release 9.5.</p> <p><b>no-prepend-global-as</b> option introduced in Junos OS Release 9.6.</p>
<b>Description</b>	<p>Specify the local autonomous system (AS) number. An AS is a set of routing devices that are under a single technical administration and generally use a single interior gateway protocol (IGP) and metrics to propagate routing information within the set of routing devices.</p> <p>Internet service providers (ISPs) sometimes acquire networks that belong to a different AS. When this occurs, there is no seamless method for moving the BGP peers of the acquired network to the AS of the acquiring ISP. The process of configuring the BGP peers with the new AS number can be time-consuming and cumbersome. In this case, it might not be desirable to modify peer arrangements or configuration. During this kind of transition period, it can be useful to configure BGP-enabled devices in the new AS to use the former AS number in BGP updates. This former AS number is called a <i>local</i> AS.</p>



**NOTE:** If you are using BGP on the routing device, you must configure an AS number before you specify the local as number.

In Junos OS Release 9.1 and later, the AS numeric range in plain-number format is extended to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*.

In Junos OS Release 9.3 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>.<16-bit low-order value in decimal>*. For

example, the 4-byte AS number of 65546 in plain-number format is represented as 1.10 in the AS-dot notation format.

**Options** **alias**—(Optional) Configure the local AS as an alias of the global AS number configured for the router at the **[edit routing-options]** hierarchy level. As a result, a BGP peer considers any local AS to which it is assigned as equivalent to the primary AS number configured for the routing device. When you use the **alias** option, only the AS (global or local) used to establish the BGP session is prepended in the AS path sent to the BGP neighbor.

**autonomous-system**—AS number.

**Range:** 1 through 4,294,967,295 ( $2^{32} - 1$ ) in plain-number format

**Range:** 0.0 through 65535.65535 in AS-dot notation format

**loops number**—(Optional) Specify the number of times detection of the AS number in the AS\_PATH attribute causes the route to be discarded or hidden. For example, if you configure **loops 1**, the route is hidden if the AS number is detected in the path one or more times. This is the default behavior. If you configure **loops 2**, the route is hidden if the AS number is detected in the path two or more times.



**NOTE:** If you configure the local AS values for any BGP group, the detection of routing loops is performed using both the AS and the local AS values for all BGP groups.

If the local AS for the EBGP or IBGP peer is the same as the current AS, do not use the **local-as** statement to specify the local AS number.

When you configure the local AS within a VRF, this impacts the AS path loop-detection mechanism. All of the **local-as** statements configured on the device are part of a single AS domain. The AS path loop-detection mechanism is based on looking for a matching AS present in the domain.

**Range:** 1 through 10

**Default:** 1

**no-prepend-global-as**—(Optional) Specify to strip the global AS and to prepend only the local AS in AS paths sent to external peers.

**private**—(Optional) Configure to use the local AS only during the establishment of the BGP session with a BGP neighbor but to hide it in the AS path sent to external BGP peers. Only the global AS is included in the AS path sent to external peers.



**NOTE:** The **private** and **alias** options are mutually exclusive. You cannot configure both options with the same **local-as** statement.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Examples: Configuring BGP Local AS on page 131</a></li><li>• <a href="#">Example: Configuring a Local AS for EBGp Sessions on page 136</a></li><li>• <i>autonomous-system</i></li><li>• <a href="#">family on page 647</a></li></ul>

---

## local-interface (IPv6)

---

<b>Syntax</b>	<code>local-interface <i>interface-name</i>;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>ipv6-link-local-address</i> ], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>ipv6-link-local-address</i> ], [edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>ipv6-link-local-address</i> ], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>ipv6-link-local-address</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Specify the interface name of the EBGp peer that uses IPv6 link-local addresses. This peer is link-local in scope.
<b>Options</b>	<i>interface-name</i> —Interface name of the EBGp IPv6 peer.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Internal BGP Peering Sessions on Logical Systems on page 68</a></li><li>• <a href="#">Example: Configuring Internal BGP Peer Sessions on page 57</a></li><li>• <a href="#">Example: Configuring External BGP on Logical Systems with IPv6 Interfaces on page 41</a></li><li>• <a href="#">Understanding Internal BGP Peering Sessions on page 56</a></li></ul>

## local-port

---

<b>Syntax</b>	<code>local-port port;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station station-name</b>],</p> <p>[edit routing-options <b>bmp</b>],</p> <p>[edit routing-options bmp <b>station station-name</b>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.3.</p>
<b>Description</b>	<p>Specifies the listening port for the BMP station connection.</p> <p>If you configure the <b>connection-mode</b> statement as <b>active</b>, do not configure the <b>local-port</b> statement. If you configure the <b>connection-mode</b> statement as <b>passive</b>, you must configure <b>local-port</b> statement.</p> <p>If you change the local port, the BMP station connection flaps when you commit the configuration.</p>
<b>Options</b>	<p><b>port</b>—Specify the local port for the BMP station connection.</p> <p><b>Range:</b> 1 through 65,535</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## local-preference

<b>Syntax</b>	<code>local-preference local-preference;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Modify the value of the <b>LOCAL_PREF</b> path attribute, which is a metric used by BGP sessions to indicate the degree of preference for an external route. The route with the highest local preference value is preferred.</p> <p>The <b>LOCAL_PREF</b> path attribute always is used in inbound routing policy and is advertised to internal BGP peers and to neighboring confederations. It is never advertised to external BGP peers.</p>
<b>Default</b>	If you omit this statement, the <b>LOCAL_PREF</b> path attribute, if present, is not modified.
<b>Options</b>	<p><b>local-preference</b>—Preference to assign to routes learned from BGP or from the group or peer.</p> <p><b>Range:</b> 0 through 4,294,967,295 (<math>2^{32} - 1</math>)</p> <p><b>Default:</b> If the <b>LOCAL_PREF</b> path attribute is present, do not modify its value. If a BGP route is received without a <b>LOCAL_PREF</b> attribute, the route is handled locally (it is stored in the routing table and advertised by BGP) as if it were received with a <b>LOCAL_PREF</b> value of 100. By default, non-BGP routes that are advertised by BGP are advertised with a <b>LOCAL_PREF</b> value of 100.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring the Local Preference Value for BGP Routes on page 80</a></li> <li>• <a href="#">Understanding Internal BGP Peering Sessions on page 56</a></li> </ul>

- [preference on page 736](#)

## log-updown (Protocols BGP)

<b>Syntax</b>	log-updown;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <a href="#">group group-name</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <a href="#">group group-name neighbor address</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <a href="#">group group-name</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name neighbor address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <a href="#">group group-name</a>],</p> <p>[edit protocols bgp group <i>group-name neighbor address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <a href="#">group group-name</a>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name neighbor address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify to generate a log a message whenever a BGP peer makes a state transition. Messages are logged using the system logging mechanism located at the <b>[edit system syslog]</b> hierarchy level.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Preventing BGP Session Resets on page 497</a></li> <li>• <i>Junos OS Administration Library for Routing Devices</i></li> <li>• <a href="#">traceoptions on page 776</a></li> </ul>

## logical-systems

---

<b>Syntax</b>	<pre>logical-systems {     logical-system-name {         ...logical-system-configuration...     } }</pre>
<b>Hierarchy Level</b>	[edit]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement name changed from <b>logical-routers</b> in Junos OS Release 9.3.
<b>Description</b>	Configure a logical system.
<b>Options</b>	<b>logical-system-name</b> —Name of the logical system.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Logical Systems Feature Guide for Routing Devices</i></li></ul>

## loops

<b>Syntax</b>	<code>loops <i>number</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp family <i>address-family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family <i>address-family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> local-as],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family <i>address-family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> local-as]</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp local-as],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options autonomous-system <i>as-number</i>],</p> <p>[edit protocols bgp family <i>address-family</i>],</p> <p>[edit protocols bgp group <i>group-name</i> family <i>address-family</i>],</p> <p>[edit protocols bgp group <i>group-name</i> local-as],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> family <i>address-family</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> local-as]</p> <p>[edit protocols bgp local-as],</p> <p>[edit routing-options autonomous-system <i>as-number</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 9.6.
<b>Description</b>	<p>Globally, for the local-AS BGP attribute, or the specified address family, allow the local device's AS number to be in the received AS paths, and specify the number of times detection of the local device's AS number in the AS_PATH attribute causes the route to be discarded or hidden. For example, if you configure <b>loops 1</b>, the route is hidden if the local device's AS number is detected in the path one or more times. This prevents routing loops and is the default behavior. If you configure <b>loops 2</b>, the route is hidden if the local device's AS number is detected in the path two or more times.</p> <p>Some examples of BGP address families are as follows:</p> <ul style="list-style-type: none"> <li>• <b>inet unicast</b></li> <li>• <b>inet-vpn multicast</b></li> <li>• <b>inet6 any</b></li> <li>• <b>l2vpn auto-discovery-only</b></li> <li>• ...</li> </ul> <p>This list is truncated for brevity. For a complete list of protocol families for which you can specify the <b>loops</b> statement, enter the <b>help apropos loops</b> configuration command at the <b>[edit protocols bgp]</b> hierarchy level on your device.</p> <pre>[edit protocols bgp] user@host# help apropos loops set family inet unicast loops     Allow local AS in received AS paths set family inet unicast loops &lt;loops&gt;     AS-Path loop count set family inet multicast loops</pre>

```

    Allow local AS in received AS paths
set family inet multicast loops <loops>
    AS-Path loop count
set family inet flow loops
    Allow local AS in received AS paths
set family inet flow loops <loops>
    AS-Path loop count
set family inet any loops
    Allow local AS in received AS paths
set family inet any loops <loops>
    AS-Path loop count
set family inet labeled-unicast loops
    Allow local AS in received AS paths
...

```



**NOTE:** When you configure the `loops` statement for a specific BGP address family, that value is used to evaluate the AS path for routes received by a BGP peer for the specified address family, rather than the `loops` value configured for the global AS number with the `loops` statement at the `[edit routing-options autonomous-system as-number]` hierarchy level.

**Options** *number*—Number of times detection of the AS number in the AS\_PATH attribute causes the route to be discarded or hidden.  
**Range:** 1 through 10  
**Default:** 1

**Required Privilege Level** routing—To view this statement in the configuration.  
 routing-control—To add this statement to the configuration.


**Related Documentation**

- [Example: Disabling Suppression of Route Advertisements on page 200](#)
- [autonomous-system](#)
- [family on page 647](#)
- [local-as on page 684](#)

## loose-check (BGP BFD Authentication)

<b>Syntax</b>	<code>loose-check ;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit protocols bgp bgp bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection authentication],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection authentication]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	<p>Specify loose authentication checking on the BFD session. Use loose authentication for transitional periods only when authentication might not be configured at both ends of the BFD session.</p> <p>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 <i>loose checking</i>. When loose checking is configured, packets are accepted without authentication being checked at each end of the session.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BFD for Static Routes</a></li> <li>• <a href="#">Example: Configuring BFD Authentication for Static Routes</a></li> <li>• <a href="#">Example: Configuring BFD on Internal BGP Peer Sessions on page 350</a></li> <li>• <a href="#">Example: Configuring BGP Route Authentication on page 449</a></li> <li>• <a href="#">Example: Configuring EBGp Multihop Sessions on page 241</a></li> </ul>

## malformed-route-limit (Protocols BGP)

<b>Syntax</b>	<code>malformed-route-limit <i>number</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <a href="#">bgp-error-tolerance</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2.
<b>Description</b>	Configure a limit on the number of malformed hidden routes stored in memory.
<div>  <p><b>NOTE:</b> When the value of <code>malformed-route-limit</code> is reduced, only new malformed BGP update messages are affected and the existing malformed routes are retained.</p> </div>	
<b>Options</b>	<p><i>number</i>—Configure a limit on the number of malformed hidden routes stored in memory.</p> <p><b>Default:</b> 1000</p> <p><b>Range:</b> 0 through 4294967295</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Error Handling for BGP Update Messages on page 525</a></li> <li>• <a href="#">Example: Configuring Error Handling for BGP Update Messages on page 527</a></li> </ul>

## malformed-update-log-interval (Protocols BGP)

<b>Syntax</b>	<code>malformed-update-log-interval seconds;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <a href="#">bgp-error-tolerance</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2.
<b>Description</b>	<p>Configure the duration for which the logging of malformed BGP update messages are suppressed.</p> <p>On configuring the malformed update log interval:</p> <ol style="list-style-type: none"> <li>1. The first malformed BGP update message is logged.</li> <li>2. All subsequent malformed update messages are suppressed until the log interval expires.</li> <li>3. On log interval expiry, the total number of malformed attributes received during the interval are logged.</li> </ol> <p>This process repeats when the next malformed update message is received.</p>
<b>Options</b>	<p><b>seconds</b>—Configure the duration for which the logging of malformed BGP update messages are suppressed.</p> <p><b>Default:</b> 300 seconds</p> <p><b>Range:</b> 10 through 65535 seconds</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Error Handling for BGP Update Messages on page 525</a></li> <li>• <a href="#">Example: Configuring Error Handling for BGP Update Messages on page 527</a></li> </ul>

## max-sessions (Origin Validation for BGP)

---

<b>Syntax</b>	<code>max-sessions <i>number</i>;</code>
<b>Hierarchy Level</b>	<code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i>],</code> <code>[edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i>],</code> <code>[edit routing-options validation group <i>group-name</i>]</code>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	<p>Configure the number of concurrent sessions for each group.</p> <p>If the number of sessions in a group exceeds the <b>max-sessions</b> value, the connections are established in order by <b>preference</b> value. A numerically higher preference results in a higher probability for session establishment. The order of session establishment is random among sessions with equal preferences.</p>
<b>Options</b>	<p><b>number</b>—Maximum number of sessions per group.</p> <p><b>Range:</b> 1 through 63</p> <p><b>Default:</b> 2</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## maximum-length (Origin Validation for BGP)

<b>Syntax</b>	<pre>maximum-length <i>prefix-length</i> {     <i>origin-autonomous-system as-number</i> {         <i>validation-state</i> (invalid   valid);     } }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation static record <i>destination</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i>],</p> <p>[edit routing-options validation static record <i>destination</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure the maximum prefix-length for a route validation (RV) record. This is a required statement.
<b>Options</b>	<p><b><i>prefix-length</i></b>—Maximum prefix-length range for a given RV entry.</p> <p><b>Range:</b> 1 through 128</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## metric-out (Protocols BGP)

<b>Syntax</b>	<code>metric-out (<i>metric</i>   minimum-igp <i>offset</i>   igp (delay-med-update   <i>offset</i>);</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name neighbor address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <i>group group-name</i>],</p> <p>[edit protocols bgp <i>group group-name neighbor address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p>Option <b>delay-med-update</b> introduced in Junos OS Release 9.0.</p>
<b>Description</b>	<p>Specify the metric for all routes sent using the multiple exit discriminator (MED, or <b>MULTI_EXIT_DISC</b>) path attribute in update messages. This path attribute is used to discriminate among multiple exit points to a neighboring AS. If all other factors are equal, the exit point with the lowest metric is preferred.</p> <p>You can specify a constant metric value by including the <b>metric</b> option. For configurations in which a BGP peer sends third-party next hops that require the local system to perform next-hop resolution—IBGP configurations, configurations within confederation peers, or EBGP configurations that include the <b>multihop</b> command—you can specify a variable metric by including the <b>minimum-igp</b> or <b>igp</b> option.</p> <p>You can increase or decrease the variable metric calculated from the IGP metric (either from the <b>igp</b> or <b>minimum-igp</b> statement) by specifying a value for <b>offset</b>. The metric is increased by specifying a positive value for <b>offset</b>, and decreased by specifying a negative value for <b>offset</b>.</p> <p>In Junos OS Release 9.0 and later, you can specify that a BGP group or peer not advertise updates for the MED path attributes used to calculate IGP costs for BGP next hops unless the MED is lower. You can also configure an interval to delay when MED updates are sent by including the <b>med-igp-update-interval minutes</b> statement at the [edit routing-options] hierarchy level.</p>
<b>Options</b>	<p><b>delay-med-update</b>—Specify that a BGP group or peer configured with the <b>metric-out igp</b> statement not advertise MED updates unless the current MED value is lower than</p>

the previously advertised MED value, or another attribute associated with the route has changed, or the BGP peer is responding to a refresh route request.



**NOTE:** You cannot configure the `delay-med-update` statement at the global BGP level.

**igp**—Set the metric to the most recent metric value calculated in the IGP to get to the BGP next hop. Routes learned from an EBGP peer usually have a next hop on a directly connected interface and thus the IGP value is equal to zero. This is the value advertised.

**metric**—Primary metric on all routes sent to peers.

**Range:** 0 through 4,294,967,295 ( $2^{32} - 1$ )

**Default:** No metric is sent.

**minimum-igp**—Set the metric to the minimum metric value calculated in the IGP to get to the BGP next hop. If a newly calculated metric is greater than the minimum metric value, the metric value remains unchanged. If a newly calculated metric is lower, the metric value is lowered to that value. When you change a neighbor's export policy from any configuration to a configuration that sets the minimum IGP offset on an exported route, the advertised MED is not updated if the value would increase as a result, even if the previous configuration does not use a minimum IGP-based MED value. This behavior helps to prevent unnecessary route flapping when an IGP cost changes, by not forcing a route update if the metric value increases past the previous lowest known value.

**offset**—Increases or decreases the metric by this value.

**Range:**  $-2^{31}$  through  $2^{31} - 1$

**Default:** None

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
---------------------------------	---

<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates on page 121</a></li> <li>• <a href="#">Examples: Configuring BGP MED on page 92</a></li> <li>• <a href="#">Example: Configuring the MED Attribute Directly on page 95</a></li> <li>• <a href="#">Understanding the MED Attribute on page 92</a></li> <li>• <code>med-igp-update-interval</code></li> </ul>
------------------------------	--

## minimum-interval (BFD Liveness Detection)

<b>Syntax</b>	<code>minimum-interval <i>milliseconds</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection],</p> <p>[edit protocols bgp bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement 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> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	<p>Configure the minimum interval after which the local routing device transmits hello packets and then expects to receive a reply from a 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 <a href="#">minimum-interval</a> (specified under the <a href="#">transmit-interval</a> statement) and <a href="#">minimum-receive-interval</a> statements.</p>
<b>Options</b>	<p><i>milliseconds</i>—Specify the minimum interval value for BFD liveliness detection.</p> <p><b>Range:</b> 1 through 255,000</p>

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation**

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- [bfd-liveness-detection on page 627](#)
- [minimum-receive-interval on page 704](#)
- [transmit-interval on page 783](#)

## minimum-interval (transmit-interval)

<b>Syntax</b>	<code>minimum-interval <i>milliseconds</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection transmit-interval]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.2.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Configure the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session. Optionally, instead of using

this statement at this hierarchy level, you can configure the minimum transmit interval using the [minimum-interval](#) statement at the **bfd-liveness-detection** hierarchy level.

**Options** *milliseconds*—Minimum transmit interval value.

**Range:** 1 through 255,000



**NOTE:** The threshold value specified in the **threshold** statement must be greater than the value specified in the **minimum-interval** statement for the **transmit-interval** statement.

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- [bfd-liveness-detection on page 627](#)
- [minimum-interval on page 700](#)
- [threshold on page 772](#)

## minimum-receive-interval (BFD Liveness Detection)

<b>Syntax</b>	<code>minimum-receive-interval <i>milliseconds</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection],</p> <p>[edit protocols bgp bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.5.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Configure the minimum interval after which the local routing device must 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 <b>minimum-interval</b> statement.
<b>Options</b>	<p><b><i>milliseconds</i></b>—Specify the minimum receive interval value.</p> <p><b>Range:</b> 1 through 255,000</p>

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- [bfd-liveness-detection on page 627](#)
- [minimum-interval on page 700](#)
- [transmit-interval on page 783](#)

## monitor (Protocols BMP)

**Syntax** monitor (enable | disable);

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols bgp [bmp](#)],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bmp],  
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bmp],  
[edit logical-systems *logical-system-name* routing-options [bmp](#)],  
[edit logical-systems *logical-system-name* routing-options bmp [station](#) *station-name*],  
[edit protocols bgp [bmp](#)],  
[edit protocols bgp group *group-name* bmp],  
[edit protocols bgp group *group-name* neighbor *address* bmp],  
[edit routing-options [bmp](#)],  
[edit routing-options bmp [station](#) *station-name*]

**Release Information** Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series.  
Statement introduced in Junos OS Release 13.3.

**Description** BMP monitoring is enabled by default. You can explicitly enable BMP monitoring or disable it. You can also selectively enable or disable BMP monitoring at various hierarchy levels (for example, [edit protocols bgp group *group-name*] or [edit protocols bgp group *group-name* neighbor *address*]). If you disable BMP monitoring, withdrawal messages are sent for any previously advertised routes. These are followed by a down message. If you enable BMP monitoring, an up message is sent first and then the route advertisements follow.

**Options** **enable**—Enable BMP monitoring.  
**Default:** BMP monitoring is enabled by default.  
**disable**—Disable BMP monitoring.

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

## mtu-discovery

<b>Syntax</b>	mtu-discovery;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Configure TCP path maximum transmission unit (MTU) discovery.</p> <p>TCP path MTU discovery enables BGP to automatically discover the best TCP path MTU for each BGP session. In Junos OS, TCP path MTU discovery is disabled by default for all BGP neighbor sessions.</p> <p>When MTU discovery is disabled, TCP sessions that are not directly connected transmit packets of 512-byte maximum segment size (MSS). These small packets minimize the chances of packet fragmentation at a device along the path to the destination. However, because most links use an MTU of at least 1500 bytes, 512-byte packets do not result in the most efficient use of link bandwidth. For directly connected EBGP sessions, MTU mismatches prevent the BGP session from being established. As a workaround, enable path MTU discovery within the EBGP group.</p> <p>Path MTU discovery dynamically determines the MTU size on the network path between the source and the destination, with the goal of avoiding IP fragmentation. Path MTU discovery works by setting the Don't Fragment (DF) bit in the IP headers of outgoing packets. When a device along the path has an MTU that is smaller than the packet, the device drops the packet. The device also sends back an ICMP Fragmentation Needed (Type 3, Code 4) message that contains the device's MTU, thus allowing the source to reduce its path MTU appropriately. The process repeats until the MTU is small enough to traverse the entire path without fragmentation.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Example: Limiting TCP Segment Size for BGP on page 468](#)
  - *Configuring the Junos OS for IPv6 Path MTU Discovery*
  - *Configuring the Junos OS for Path MTU Discovery on Outgoing GRE Tunnel Connections*

## multihop

**Syntax**    `multihop {  
                  no-nexthop-change;  
                  ttl ttl-value;  
                  }`

**Hierarchy Level**    [edit logical-systems *logical-system-name* protocols bgp],  
[edit logical-systems *logical-system-name* protocols bgp **group** *group-name*],  
[edit logical-systems *logical-system-name* protocols bgp **group** *group-name*  
                  **neighbor** *address*],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols  
                  bgp],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols  
                  bgp **group** *group-name*],  
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols  
                  bgp group *group-name* **neighbor** *address*],  
[edit protocols bgp],  
[edit protocols bgp **group** *group-name*],  
[edit protocols bgp group *group-name* **neighbor** *address*],  
[edit routing-instances *routing-instance-name* protocols bgp],  
[edit routing-instances *routing-instance-name* protocols bgp **group** *group-name*],  
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*  
                  **neighbor** *address*]

**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 11.3 for the QFX Series.

**Description**    Configure an EBGp multihop session.

For Layer 3 VPNs, you configure the EBGp multihop session between the PE and CE routing devices. This allows you to configure one or more routing devices between the PE and CE routing devices.

An external confederation peer is a special case that allows unconnected third-party next hops. You do not need to configure multihop sessions explicitly in this particular case because multihop behavior is implied.

If you have external BGP confederation peer-to-loopback addresses, you still need the multihop configuration.



**NOTE:** You cannot configure the `accept-remote-nexthop` statement at the same time.

**Default**    If you omit this statement, all EBGp peers are assumed to be directly connected (that is, you are establishing a nonmultihop, or “regular,” BGP session), and the default time-to-live (TTL) value is 1.

The remaining statements are explained separately.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring EBGp Multihop Sessions on page 241</a></li><li>• <i>Configuring EBGp Multihop Sessions Between PE and CE Routers in Layer 3 VPNs</i></li><li>• <a href="#">accept-remote-nextthop on page 604</a></li><li>• <a href="#">no-nextthop-change on page 721</a></li><li>• <a href="#">ttl on page 785</a></li></ul>

## multipath (Protocols BGP)

<b>Syntax</b>	<pre> multipath {   multiple-as;   vpn-unequal-cost equal-external-internal; } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Allow load sharing among multiple EBGP paths and multiple IBGP paths. A path is considered a BGP equal-cost path (and will be used for forwarding) if a tie-break is performed. The tie-break is performed after the BGP route path selection step that chooses the next-hop path that is resolved through the IGP route with the lowest metric. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor, are considered.</p>
<b>Options</b>	<p><b>multiple-as</b>—Disable the default check requiring that paths accepted by BGP multipath must have the same neighboring AS.</p> <p><b>vpn-unequal-cost equal-external-internal</b>—Enable load-balancing in a Layer 3 VPN with unequal cost paths.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding BGP Path Selection on page 8</a></li> <li>• <a href="#">Example: Load Balancing BGP Traffic on page 366</a></li> </ul>

## multiplier (BFD Liveness Detection)

<b>Syntax</b>	<code>multiplier <i>number</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection],</p> <p>[edit protocols bgp bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.5.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.
<b>Options</b>	<p><i>number</i>—Number of hello packets.</p> <p><b>Range:</b> 1 through 255</p> <p><b>Default:</b> 3</p>

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- [bfd-liveness-detection on page 627](#)

## neighbor (Protocols BGP)

```
Syntax  neighbor address {
    accept-remote-nexthop;
    advertise-external <conditional>;
    advertise-inactive;
    (advertise-peer-as | no-advertise-peer-as);
    as-override;
    authentication-algorithm algorithm;
    authentication-key key;
    authentication-key-chain key-chain;
    cluster cluster-identifier;
    damping;
    description text-description;
    export [ policy-names ];
    family {
        (inet | inet6 | inet-mvpn | inet6-mvpn | inet-vpn | inet6-vpn | iso-vpn | l2-vpn) {
            (any | flow | multicast | unicast | signaling) {
                accepted-prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                damping;
                prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                rib-group group-name;
                topology name {
                    community {
                        target identifier;
                    }
                }
            }
        }
        flow {
            no-validate policy-name;
        }
        labeled-unicast {
            accepted-prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            aggregate-label {
                community community-name;
            }
            explicit-null {
                connected-only;
            }
            prefix-limit {
                maximum number;
                teardown <percentage> <idle-timeout (forever | minutes)>;
            }
            resolve-vpn;
            rib inet.3;
        }
    }
}
```

```

    rib-group group-name;
    topology name {
        community {
            target identifier;
        }
    }
}
}
route-target {
    advertise-default;
    external-paths number;
    accepted-prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
    }
}
signaling {
    prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
    }
}
}
graceful-restart {
    disable;
    restart-time seconds;
    stale-routes-time seconds;
}
hold-time seconds;
import [ policy-names ];
ipsec-sa ipsec-sa;
keep (all | none);
local-address address;
local-as autonomous-system <private>;
local-interface interface-name;
local-preference preference;
log-updown;
metric-out (metric | minimum-igp <offset> | igp <offset>);
mtu-discovery;
multihop <ttl-value>;
multipath {
    multiple-as;
}
no-aggregator-id;
no-client-reflect;
out-delay seconds;
passive;
peer-as autonomous-system;
preference preference;
tcp-aggressive-transmission;
tcp-mss segment-size;
traceoptions {

```

```

    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  vpn-apply-export;
}

```

Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> ], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> ], [edit protocols bgp <b>group</b> <i>group-name</i> ], [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <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. Statement introduced in Junos OS Release 11.3 for the QFX Series.
Description	<p>Explicitly configure a neighbor (peer). To configure multiple BGP peers, include multiple <b>neighbor</b> statements.</p> <p>By default, the peer's options are identical to those of the group. You can override these options by including peer-specific option statements within the <b>neighbor</b> statement.</p> <p>The <b>neighbor</b> statement is one of the statements you can include in the configuration to define a minimal BGP configuration on the routing device. (You can include an <b>allow all</b> statement in place of a <b>neighbor</b> statement.)</p>
Options	<p><b>address</b>—IPv6 or IPv4 address of a single peer.</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"> <li>• <i>BGP Feature Guide for Routing Devices</i></li> </ul>

## no-adaptation (BFD Liveness Detection)

<b>Syntax</b>	no-adaptation;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection],</p> <p>[edit protocols bgp bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Configure BFD sessions not to adapt to changing network conditions. We recommend that you <i>do not</i> disable BFD adaptation unless it is preferable to have BFD adaptation disabled in your network.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>Configuring BFD for Layer 2 VPN and VPLS</li> </ul>

- [Example: Configuring BFD for Static Routes](#)
- [bfd-liveness-detection on page 627](#)

## no advertise-peer-as

<b>Syntax</b>	no-advertise-peer-as;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Enable the default behavior of suppressing AS routes.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BGP Route Advertisement on page 233</a></li> <li>• <a href="#">Understanding Route Advertisement on page 233</a></li> <li>• <a href="#">advertise-peer-as on page 612</a></li> </ul>

## no-aggregator-id

<b>Syntax</b>	no-aggregator-id;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Prevent different routing devices within an AS from creating aggregate routes that contain different AS paths.</p> <p>Junos OS performs route aggregation, which is the process of combining the characteristics of different routes so that only a single route is advertised. Aggregation reduces the amount of information that BGP must store and exchange with other BGP systems. When aggregation occurs, the local routing device adds the local AS number and the router ID to the aggregator path attribute. The <b>no-aggregator-id</b> statement causes Junos OS to place a 0 in the router ID field and thus eliminate the possibility of having multiple aggregate advertisements in the network, each with different path information.</p>
<b>Default</b>	If you omit this statement, the router ID is included in the BGP aggregator path attribute.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Update Messages on page 8</a></li> </ul>

## no-client-reflect


<b>Syntax</b>	no-client-reflect;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],  [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],  [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],  [edit protocols bgp],  [edit protocols bgp <b>group</b> <i>group-name</i>],  [edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],  [edit routing-instances <i>routing-instance-name</i> protocols bgp],  [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Disable intracluster route redistribution by the system acting as the route reflector. Include this statement when the client cluster is fully meshed to prevent the sending of redundant route advertisements. Route reflection provides a way to decrease BGP control traffic and minimizing the number of update messages sent within the AS.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BGP Route Reflectors on page 419</a></li> <li>• <a href="#">cluster on page 636</a></li> </ul>

## no-malformed-route-limit (Protocols BGP)

---

<b>Syntax</b>	no-malformed-route-limit;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp <a href="#">bgp-error-tolerance</a> ], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a> ], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a> ], [edit protocols bgp <a href="#">bgp-error-tolerance</a> ], [edit protocols bgp group <i>group-name</i> <a href="#">bgp-error-tolerance</a> ], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <a href="#">bgp-error-tolerance</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2.
<b>Description</b>	Disable the limit on the number of malformed hidden routes stored in memory.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Error Handling for BGP Update Messages on page 525</a></li><li>• <a href="#">Example: Configuring Error Handling for BGP Update Messages on page 527</a></li></ul>

## no-nexthop-change (BGP multihop)

<b>Syntax</b>	<code>no-nexthop-change; no-nexthop-self</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp multihop],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> multihop],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> multihop],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp multihop],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> multihop],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> multihop],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> multihop],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> multihop],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp multihop],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> multihop],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i> multihop]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify that the BGP next-hop value not be changed. For route advertisements, specify the <b>no-nexthop-self</b> option.</p> <p>An external confederation peer is a special case that allows unconnected third-party next hops. You do not need to configure multihop sessions explicitly in this particular case; multihop behavior is implied.</p> <p>If you have external BGP confederation peer-to-loopback addresses, you still need the multihop configuration.</p>
	<div>  <p><b>NOTE:</b> You cannot configure the <b>accept-remote-nexthop</b> statement at the same time.</p> </div>
<b>Default</b>	If you omit this statement, all EBGP peers are assumed to be directly connected (that is, you are establishing a nonmultihop, or “regular,” BGP session), and the default time-to-live (TTL) value is 1.
<b>Options</b>	<b>no-nexthop-self</b> — Specify ....
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Example: Configuring EBGp Multihop Sessions on page 241](#)
  - [accept-remote-nextthop on page 604](#)
  - [ttl on page 785](#)

## no-validate

<b>Syntax</b>	<code>no-validate <i>policy-name</i>;</code>
<b>Hierarchy Level</b>	<code>[edit protocols bgp group <i>group-name</i> <i>family</i> (inet   inet flow)],</code> <code>[edit protocols bgp group <i>group-name</i> neighbor address <i>family</i> (inet   inet flow)],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <i>family</i> (inet   inet flow)],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor address <i>family</i> (inet   inet flow)]</code>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>When BGP is carrying flow-specification network layer reachability information (NLRI) messages, the <b>no-validate</b> statement omits the flow route validation procedure after packets are accepted by a policy.</p> <p>The receiving BGP-enabled device accepts a flow route if it passes the following criteria:</p> <ul style="list-style-type: none"> <li>• The originator of a flow route matches the originator of the best match unicast route for the destination address that is embedded in the route.</li> <li>• There are no more specific unicast routes, when compared to the destination address of the flow route, for which the active route has been received from a different next-hop autonomous system.</li> </ul> <p>The first criterion ensures that the filter is being advertised by the next-hop used by unicast forwarding for the destination address embedded in the flow route. For example, if a flow route is given as 10.1.1.1, proto=6, port=80, the receiving BGP-enabled device selects the more specific unicast route in the unicast routing table that matches the destination prefix 10.1.1.1/32. On a unicast routing table containing 10.1/16 and 10.1.1/24, the latter is chosen as the unicast route to compare against. Only the active unicast route entry is considered. This follows the concept that a flow route is valid if advertised by the originator of the best unicast route.</p> <p>The second criterion addresses situations in which a given address block is allocated to different entities. Flows that resolve to a best-match unicast route that is an aggregate route are only accepted if they do not cover more specific routes that are being routed to different next-hop autonomous systems.</p> <p>You can bypass the validation process and use your own specific import policy. To disable the validation procedure and use an import policy instead, include the <b>no-validate</b> statement in the configuration.</p> <p>Flow routes configured for VPNs with family <b>inet-vpn</b> are not automatically validated, so the <b>no-validate</b> statement is not supported at the <code>[edit protocols bgp group <i>group-name</i> family inet-vpn]</code> hierarchy level. No validation is needed if the flow routes are configured locally between devices in a single AS.</p>

<b>Options</b>	<i>policy-name</i> —Import policy to match NLRI messages.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Flow Routes on page 550</a></li></ul>

---

## origin-autonomous-system (Origin Validation for BGP)

---

<b>Syntax</b>	<pre>origin-autonomous-system <i>as-number</i> {     <i>validation-state</i> (invalid   valid); }</pre>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> ], [edit logical-systems <i>logical-system-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> ], [edit routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> ], [edit routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure the legitimate originator autonomous system (AS). This is a required statement.
<b>Options</b>	<i>as-number</i> —Legitimate originator AS number.  The remaining statement is explained separately.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## out-delay

<b>Syntax</b>	<code>out-delay seconds;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit protocols bgp],          [edit protocols bgp <b>group</b> <i>group-name</i>],          [edit protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Control how often BGP and the routing table exchange route information by specifying how long a route must be present in the Junos OS routing table before it is exported to BGP. Use this time delay to help bundle routing updates and to avoid sending updates too often.</p> <p>Alternatively or in addition, external BGP (EBGP) sessions can also use the route-flap damping mechanism upon the reception of BGP messages coming from an external neighbor.</p> <p>BGP stores the route information it receives from update messages in the routing table, and the routing table exports active routes from the routing table into BGP. BGP then advertises the exported routes to its peers. The <b>out-delay</b> statement enables a form of rate limiting. The delay is added to each update for each prefix individually. When a routing device changes its best path to a destination prefix, the device does not inform its peer about the change unless the route has been present in its routing table for the specified <b>out-delay</b>. If you use <b>out-delay</b> to perform rate-limiting, you can expect a less bursty pattern of updates. You will see a pattern in which updates arrive in a steady flow, and two updates for the same prefix are always spaced by at least the <b>out-delay</b> timer value (for example, 30 seconds). Thus, the <b>out-delay</b> setting is useful for limiting oscillation (sometimes called <i>churn</i>) in a network. Keep in mind that, regardless of the <b>out-delay</b> setting, BGP peers exchange routes immediately after neighbor establishment. The <b>out-delay</b> setting is only designed to delay the exchange of routes between BGP and the local routing table.</p>

Caution is warranted because an **out-delay** can delay convergence. If your network is configured in a way that avoids oscillation, setting an **out-delay** is not necessary.

When configured, the **out-delay** value displays as **Outbound Timer** when using **show bgp group** or **show bgp group neighbor** commands.


**Default** By default, the exchange of route information between BGP and the routing table occurs immediately after the routes are received. This immediate exchange of route information might cause instabilities in the network reachability information. If you omit this statement, routes are exported to BGP immediately after they have been added to the routing table.

**Options** *seconds*—Output delay time.  
**Range:** 0 through 65,535 seconds  
**Default:** 0 seconds

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation** • [keep on page 674](#)

## outbound-route-filter


<b>Syntax</b>	<pre> outbound-route-filter {     <b>bgp-orf-cisco-mode</b>;     prefix-based {         accept {             (inet   inet6);         }     } } </pre>
<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> neighbor <i>address</i>], [edit protocols bgp], [edit protocols bgp group <i>group-name</i>], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor   <i>address</i>] </pre>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.2.</p> <p>Statement introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p>
<b>Description</b>	Configure a BGP peer to accept outbound route filters from a remote peer.
<b>Options</b>	<p><b>accept</b>—Specify that outbound route filters from a BGP peer be accepted.</p> <p><b>inet</b>—Specify that IPv4 prefix-based outbound route filters be accepted.</p> <p><b>inet6</b>—Specify that IPv6 prefix-based outbound route filters be accepted.</p>
	<p> <b>NOTE:</b> You can specify that both IPv4 and IPv6 outbound route filters be accepted.</p>
	<p><b>prefix-based</b>—Specify that prefix-based filters be accepted.</p> <p>The <b>bgp-orf-cisco-mode</b> statement is explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 237](#)

## passive (Protocols BGP)

<b>Syntax</b>	passive;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group</i> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit protocols bgp],          [edit protocols bgp <b>group</b> <i>group-name</i>],          [edit protocols bgp <i>group</i> <i>group-name</i> <b>neighbor</b> <i>address</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],          [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group</i> <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Configure the routing device so that active open messages are not sent to the peer. Once you configure the routing device to be passive, the routing device will wait for the peer to issue an open request before a message is sent.
<b>Default</b>	If you omit this statement, all explicitly configured peers are active, and each peer periodically sends open requests until its peer responds.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Preventing BGP Session Flaps When VPN Families Are Configured on page 497</a></li> </ul>

## path-count

<b>Syntax</b>	<code>path-count <i>number</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family <i>family</i> add-path send],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor address family <i>family</i> add-path send],</p> <p>[edit protocols bgp group <i>group-name</i> family <i>family</i> add-path send],</p> <p>[edit protocols bgp group <i>group-name</i> family <i>family</i> add-path neighbor address family <i>family</i> add-path send]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 11.3.</p> <p>Support for range from 2 through 20 (for BGP) introduced in Junos OS Release 14.1.</p>
<b>Description</b>	<p>Specify the number of paths to a destination to advertise.</p> <p>Suppose a routing device has in its routing table four paths to a destination and is configured to advertise up to three paths (<b>add-path send path-count 3</b>). The three paths are chosen based on path selection criteria. That is, the three best paths are chosen in path-selection order. The best path is the active path. This path is removed from consideration and a new best path is chosen. This process is repeated until the specified number of paths is reached.</p>
<b>Options</b>	<p><b>number</b>—Number of paths to a destination to advertise.</p> <p><b>Range:</b> 2 through 6</p> <p><b>Default:</b> 1</p> <p><b>Range:</b> 2 through 20 (for BGP)</p>
<div>  <p><b>NOTE:</b> This range is applicable only under <b>prefix-policy add-path</b>.</p> </div>	
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Advertising Multiple BGP Paths to a Destination on page 392</a></li> <li>• <a href="#">prefix-policy on page 740</a></li> </ul>

## path-selection

<b>Syntax</b>	<pre>path-selection {   (always-compare-med   cisco-non-deterministic   external-router-id);   as-path-ignore;   l2vpn-use-bgp-rules;   med-plus-igp {     igp-multiplier <i>number</i>;     med-multiplier <i>number</i>;   } }</pre>
<b>Hierarchy Level</b>	<pre>[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp]</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>med-plus-igp</b> option introduced in Junos OS Release 8.1.</p> <p><b>as-path-ignore</b> and <b>l2vpn-use-bgp-rules</b> options introduced in Junos OS Release 10.2.</p>
<b>Description</b>	Configure BGP path selection.
<b>Default</b>	If the <b>path-selection</b> statement is not included in the configuration, only the multiple exit discriminators (MEDs) of routes that have the same peer ASs are compared.
<b>Options</b>	<b>always-compare-med</b> —Always compare MEDs whether or not the peer ASs of the compared routes are the same.



**NOTE:** We recommend that you configure the **always-compare-med** option.

**as-path-ignore**—In the best-path algorithm, skip the step that compares the autonomous system (AS) path lengths. By default, the best-path algorithm evaluates the length of the AS paths and prefers the route with the shortest AS path length.



**NOTE:** The **as-path-ignore** statement is not supported with routing instances.

**cisco-non-deterministic**—Emulate the Cisco IOS default behavior. This mode evaluates routes in the order that they are received and does not group them according to their neighboring AS. With **cisco-non-deterministic** mode, the active path is always first. All inactive, but eligible, paths follow the active path and are maintained in the order

in which they were received, with the most recent path first. Ineligible paths remain at the end of the list.

As an example, suppose you have three path advertisements for the 192.168.1.0 /24 route:

- Path 1—learned through EBGp; AS Path of 65010; MED of 200
- Path 2—learned through IBGP; AS Path of 65020; MED of 150; IGP cost of 5
- Path 3—learned through IBGP; AS Path of 65010; MED of 100; IGP cost of 10

These advertisements are received in quick succession, within a second, in the order listed. Path 3 is received most recently, so the routing device compares it against path 2, the next most recent advertisement. The cost to the IBGP peer is better for path 2, so the routing device eliminates path 3 from contention. When comparing paths 1 and 2, the routing device prefers path 1 because it is received from an EBGp peer. This allows the routing device to install path 1 as the active path for the route.



**NOTE:** We do not recommend using this configuration option in your network. It is provided solely for interoperability to allow all routing devices in the network to make consistent route selections.

**external-router-id**—Compare the router ID between external BGP paths to determine the active path.

**igp-multiplier *number***—The multiplier value for the IGP cost to a next-hop address. This option is useful for making the MED and IGP cost comparable.

**Range:** 1 through 1000

**Default:** 1

**med-multiplier *number***—The multiplier value for the MED calculation. This option is useful for making the MED and IGP cost comparable.

**Range:** 1 through 1000

**Default:** 1

**med-plus-igp**—Add the IGP cost to the indirect next-hop destination to the MED before comparing MED values for path selection. This statement only affects best-path selection. It does not affect the advertised MED.

The other option 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**

- [Understanding BGP Path Selection on page 8](#)
- [Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 260](#)

## peer-as (Protocols BGP)

<b>Syntax</b>	<code>peer-as <i>autonomous-system</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify the neighbor (peer) autonomous system (AS) number.</p> <p>For EBGP, the peer is in another AS, so the AS number you specify in the <b>peer-as</b> statement must be different from the local router's AS number, which you specify in the <b>autonomous-system</b> statement. For IBGP, the peer is in the same AS, so the two AS numbers that you specify in the <b>autonomous-system</b> and <b>peer-as</b> statements must be the same.</p> <p>The AS numeric range in plain-number format has been extended in Junos OS Release 9.1 to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, <i>BGP Support for Four-octet AS Number Space</i>. RFC 4893 introduces two new optional transitive BGP attributes, AS4_PATH and AS4_AGGREGATOR. These new attributes are used to propagate 4-byte AS path information across BGP speakers that do not support 4-byte AS numbers. RFC 4893 also introduces a reserved, well-known, 2-byte AS number, AS 23456. This reserved AS number is called AS_TRANS in RFC 4893. All releases of the Junos OS support 2-byte AS numbers.</p> <p>In Junos OS Release 9.2 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: <i>&lt;16-bit high-order value in decimal&gt;.&lt;16-bit low-order value in decimal&gt;</i>. For example, the 4-byte AS number of 65,546 in plain-number format is represented as 1.10 in the AS-dot notation format.</p> <p>With the introduction of 4-byte AS numbers, you might have a combination of routers that support 4-byte AS numbers and 2-byte AS numbers. For more information about what happens when establishing BGP peer relationships between 4-byte and 2-byte capable routers, see the following topics:</p>

- *Using 4-Byte Autonomous System Numbers in BGP Networks Technology Overview.*

**Options** *autonomous-system*—AS number.  
**Range:** 1 through 4,294,967,295 ( $2^{32} - 1$ ) in plain-number format for 4-byte AS numbers  
**Range:** 1 through 65,535 in plain-number format for 2-byte AS numbers (this is a subset of the 4-byte range)  
**Range:** 0.0 through 65535.65535 in AS-dot notation format for 4-byte AS numbers

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation**

## port (Origin Validation for BGP)

**Syntax** *port port-number;*

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *instance-name* routing-options validation group *group-name* session *server-ip-address*],  
 [edit logical-systems *logical-system-name* routing-options validation group *group-name* session *server-ip-address*],  
 [edit routing-instances *instance-name* routing-options validation group *group-name* session *server-ip-address*],  
 [edit routing-options validation group *group-name* session *server-ip-address*]

**Release Information** Statement introduced in Junos OS Release 12.2.  
 Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**Description** Configure an alternative TCP port number to be used for the outgoing connection with the cache server. The well-known resource public key infrastructure (RPKI) port is TCP port 2222. For a given deployment, an RPKI cache server might listen on some other TCP port number. If so, configure the alternative port number with this statement.

**Options** *port-number*—TCP port number to be used for the outgoing connection to the cache server.  
**Default:** 2222

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation** • [Example: Configuring Origin Validation for BGP on page 473](#)

## pre-policy

---

<b>Syntax</b>	<pre>pre-policy {     exclude-non-feasible; }</pre>
<b>Hierarchy Level</b>	[edit protocols bgp bmp <a href="#">route-monitoring</a> ], [edit protocols bgp group <i>group-name</i> bmp <a href="#">route-monitoring</a> ], [edit protocols bgp group neighbor <i>group-name</i> neighbor <i>address</i> bmp <a href="#">route-monitoring</a> ], [edit routing-options bmp <a href="#">route-monitoring</a> ], [edit routing-options bmp station <i>station-name</i> <a href="#">route-monitoring</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Excludes routes that are non-feasible from the BMP route monitoring decision process (for example, a route loop). This represents the view of the BGP routes before running the import policy.
<b>Options</b>	<b>exclude-non-feasible</b> —Exclude routes that are non-feasible for the decision process.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li></ul>

## precision-timers

<b>Syntax</b>	precision-timers;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit protocols bgp]
<b>Release Information</b>	Statement introduced in Junos OS Release 11.4.
<b>Description</b>	Enable BGP sessions to send frequent keepalive messages with a hold time as short as 10 seconds.




**NOTE:** The hold time is three times the interval at which keepalive messages are sent, and the hold time is the maximum number of seconds allowed to elapse between successive keepalive messages that BGP receives from a peer. When establishing a BGP connection with the local routing device, a peer sends an open message, which contains a hold-time value. BGP on the local routing device uses the smaller of either the local hold-time value or the peer's hold-time value as the hold time for the BGP connection between the two peers.

The default hold-time is 90 seconds, meaning that the default frequency for keepalive messages is 30 seconds. More frequent keepalive messages and shorter hold times might be desirable in large-scale deployments with many active sessions (such as edge or large VPN deployments). To configure the hold time and the frequency of keepalive messages, include the `hold-time` statement at the [edit protocols bgp] hierarchy level. You can configure the hold time at a logical system, routing instance, global, group, or neighbor level. When you set a hold time value to less than 20 seconds, we recommend that you also configure the BGP `precision-timers` statement. The `precision-timers` statement ensures that if scheduler slip messages occur, the routing device continues to send keepalive messages. When the `precision-timers` statement is included, keepalive message generation is performed in a dedicated kernel thread, which helps to prevent BGP session flaps.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">hold-time on page 663</a></li> </ul>

## preference (Protocols BGP)

<b>Syntax</b>	<code>preference preference;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify the preference for routes learned from BGP.</p> <p>At the BGP global level, the preference statement sets the preference for routes learned from BGP. You can override this preference in a BGP group or peer preference statement.</p> <p>At the group or peer level, the preference statement sets the preference for routes learned from the group or peer. Use this statement to override the preference set in the BGP global preference statement when you want to favor routes from one group or peer over those of another.</p>
<div>  <b>NOTE:</b> Do not set preference2 for BGP route-policy.         </div>	
<b>Options</b>	<p><b>preference</b>—Preference to assign to routes learned from BGP or from the group or peer.</p> <p><b>Range:</b> 0 through 4,294,967,295 (<math>2^{32} - 1</math>)</p> <p><b>Default:</b> 170 for the primary preference</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">local-preference on page 688</a></li> <li>• <a href="#">Example: Configuring the Preference Value for BGP Routes on page 252</a></li> </ul>

## preference (Origin Validation for BGP)

<b>Syntax</b>	<code>preference <i>number</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-options validation group <i>group-name</i> session <i>server-ip-address</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	<p>Each resource public key infrastructure (RPKI) cache server has a static preference. Higher preferences are preferred. During a session start or restart, the routing device attempts to start a session with the cache server that has the numerically highest preference. The routing device connects to multiple cache servers in preference order.</p>
<b>Options</b>	<p><i>number</i>—Preference number for the cache server.</p> <p><b>Range:</b> 1 through 255</p> <p><b>Default:</b> 100</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## prefix-limit

<b>Syntax</b>	<pre>prefix-limit {     maximum <i>number</i>;     teardown &lt;<i>percentage</i>&gt; &lt;idle-timeout (forever   <i>minutes</i>)&gt;; }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit protocols bgp <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) (any   labeled-unicast   multicast   unicast)],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) (any   flow   labeled-unicast   multicast   unicast)]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Limit the number of prefixes received on a BGP peer session and a rate-limit logging when injected prefixes exceed a set limit.</p> <p>This functionality is identical to the <b>accepted-prefix-limit</b> functionality except that it operates against received prefixes rather than accepted prefixes.</p>
<b>Options</b>	<p><b>maximum <i>number</i></b>—When you set the maximum number of prefixes, a message is logged when that number is exceeded.</p> <p><b>Range:</b> 1 through 4,294,967,295 (<math>2^{32} - 1</math>)</p> <p><b>teardown &lt;<i>percentage</i>&gt;</b>—If you include the <b>teardown</b> statement, the session is torn down when the maximum number of prefixes is reached. If you specify a percentage, messages are logged when the number of prefixes exceeds that percentage. After the session is torn down, it is reestablished in a short time unless you include the <b>idle-timeout</b> statement. Then the session can be kept down for a specified amount</p>

of time, or forever. If you specify **forever**, the session is reestablished only after you issue a **clear bgp neighbor** command.

**Range:** 1 through 100

**idle-timeout (*forever* | *timeout-in-minutes*)**—(Optional) If you include the **idle-timeout** statement, the session is torn down for a specified amount of time, or forever. If you specify a period of time, the session is allowed to reestablish after this timeout period. If you specify **forever**, the session is reestablished only after you intervene with a **clear bgp neighbor** command.

**Range:** 1 through 2400

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.


<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">accepted-prefix-limit on page 605</a></li><li>• <a href="#">Understanding Multiprotocol BGP on page 537</a></li></ul>
------------------------------	---

## prefix-policy

---

<b>Syntax</b>	<code>prefix-policy [ <i>policy-names</i> ];</code>
<b>Hierarchy Level</b>	<code>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family <i>family</i> add-path send],</code> <code>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family <i>family</i> add-path send],</code> <code>[edit protocols bgp group <i>group-name</i> family <i>family</i> add-path send],</code> <code>[edit protocols bgp group <i>group-name</i> family <i>family</i> add-path neighbor <i>address</i> family <i>family</i> add-path send]</code>
<b>Release Information</b>	Statement introduced in Junos OS Release 11.3.
<b>Description</b>	<p>Filter the paths to a destination to advertise.</p> <p>Prefix policies enable you to filter routes on a router that is configured to advertise multiple paths to a destination. Prefix policies can only match prefixes. They cannot match route attributes, and they cannot change the attributes of routes.</p> <p>The <b>add-path prefix-policy</b> allows up to 20 BGP add-paths be advertised for a subset of prefixes that match the add-path prefix-policy. To enable this feature for a prefix, the <b>add-path prefix-policy</b> term matching the prefix should have a new <i>then</i> action to set <b>add-path send-count &lt;2...20&gt;</b>. This new action is not applicable if the policy-statement containing it is used in any place other than <b>add-path prefix-policy</b>.</p>
<b>Options</b>	<b><i>policy-names</i></b> —Name of a policy (or a set of policies) configured at the <code>[edit policy-options]</code> hierarchy level.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Advertising Multiple BGP Paths to a Destination on page 392</a></li><li>• <i>Actions in Routing Policy Terms</i></li></ul>

## priority (Protocols BMP)

<b>Syntax</b>	priority (high   medium   low);
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station station-name</b> ], [edit routing-options <b>bmp</b> ], [edit routing-options bmp <b>station station-name</b> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Specifies the dispatch priority for BMP. The dispatch priority controls the frequency with which the device is able to forward BMP messages to BMP stations.
<b>Options</b>	<p><b>high</b>—Specifies that the routing protocol process handle BMP requests with high urgency.</p> <p><b>medium</b>—Specifies that the routing protocol process handle BMP requests with medium urgency.</p> <p><b>low</b>—Specifies that the routing protocol process handle BMP requests with low urgency.</p> <p><b>Default:</b> The default dispatch priority is <b>low</b> to minimize interference with other routing protocol process priorities and to match the behavior of previous versions of BMP.</p>
<div>  <p><b>NOTE:</b> Setting high or medium priority may reduce the performance of the routing protocol process in its handling route convergence or other work.</p> </div>	
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## post-policy

---

<b>Syntax</b>	<code>post-policy {     exclude-non-eligible; }</code>
<b>Hierarchy Level</b>	[edit protocols bgp bmp <a href="#">route-monitoring</a> ], [edit protocols bgp group <i>group-name</i> bmp route-monitoring], [edit protocols bgp group neighbor <i>group-name</i> neighbor <i>address</i> bmp route-monitoring], [edit routing-options bmp route-monitoring], [edit routing-options bmp station <i>station-name</i> route-monitoring]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	For BMP route monitoring, allows you to excludes routes that are non-eligible for the decision process (for example, protocol nexthop not resolved). This represents the view of the BGP routes after running the import policy. If the import policy has rejected the BGP route, the route does not exist in the post policy view.
<b>Options</b>	<b>exclude-non-eligible</b> —Exclude routes that are non-eligible for the decision process.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li></ul>

## protection (Protocols BGP)

---

<b>Syntax</b>	<code>protection;</code>
<b>Hierarchy Level</b>	[edit routing-instances <i>instance-name</i> protocols bgp <a href="#">family</a> inet unicast], [edit routing-instances <i>instance-name</i> protocols bgp <a href="#">family</a> inet6 unicast], [edit routing-instances <i>instance-name</i> protocols bgp <a href="#">family</a> inet labeled-unicast], [edit routing-instances <i>instance-name</i> protocols bgp <a href="#">family</a> inet6 labeled-unicast]
<b>Description</b>	Configure the backup path to protect the active provider edge path in a Layer 3 VPN or a BGP labeled unicast path.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Configuring Provider Edge Link Protection in Layer 3 VPNs</i></li><li>• <i>Example: Configuring Provider Edge Link Protection for BGP Labeled Unicast Paths</i></li></ul>

## protection (Protocols MPLS)

---

<b>Syntax</b>	protection;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family inet <a href="#">labeled-unicast</a> ], [edit protocols bgp group <i>group-name</i> family inet <a href="#">labeled-unicast</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2.
<b>Description</b>	Configure protection on a link between two routers in different autonomous systems.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Understanding MPLS Inter-AS Link Protection</i></li></ul>

## protocols

```
Syntax protocols {
    bgp {
        ... bgp-configuration ...
    }
    isis {
        ... isis-configuration ...
    }
    ldp {
        ... ldp-configuration ...
    }
    msdp {
        ... msdp-configuration ...
    }
    mstp {
        ... mstp-configuration ...
    }
    ospf {
        domain-id domain-id;
        domain-vpn-tag number;
        route-type-community (iana | vendor);
        ... ospf-configuration ...
    }
    ospf3 {
        domain-id domain-id;
        domain-vpn-tag number;
        route-type-community (iana | vendor);
        ... ospf3-configuration ...
    }
    pim {
        ... pim-configuration ...
    }
    rip {
        ... rip-configuration ...
    }
    ripng {
        ... ripng-configuration ...
    }
    rstp {
        rstp-configuration;
    }
    vstp {
        vstp configuration;
    }
    vpls {
        vpls configuration;
    }
}
```

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],  
[edit routing-instances *routing-instance-name*]

**Release Information** Statement introduced before Junos OS Release 7.4.

Support for RIPvng introduced in Junos OS Release 9.0.  
 Statement introduced in Junos OS Release 11.1 for EX Series switches.  
 Statement introduced in Junos OS Release 11.3 for the QFX Series.

**Description** Specify the protocol for a routing instance. You can configure multiple instances of many protocol types. Not all protocols are supported on the switches. See the switch CLI.

**Options** **bgp**—Specify BGP as the protocol for a routing instance.

**isis**—Specify IS-IS as the protocol for a routing instance.

**ldp**—Specify LDP as the protocol for a routing instance.

**l2vpn**—Specify Layer 2 VPN as the protocol for a routing instance.

**msdp**—Specify the Multicast Source Discovery Protocol (MSDP) for a routing instance.

**mstp**—Specify the Multiple Spanning Tree Protocol (MSTP) for a virtual switch routing instance.

**ospf**—Specify OSPF as the protocol for a routing instance.

**ospf3**—Specify OSPF version 3 (OSPFv3) as the protocol for a routing instance.



**NOTE:** OSPFv3 supports the **no-forwarding**, **virtual-router**, and **vrf** routing instance types only.

**pim**—Specify the Protocol Independent Multicast (PIM) protocol for a routing instance.

**rip**—Specify RIP as the protocol for a routing instance.

**ripng**—Specify RIP next generation (RIPvng) as the protocol for a routing instance.

**rstp**—Specify the Rapid Spanning Tree Protocol (RSTP) for a virtual switch routing instance.

**vstp**—Specify the VLAN Spanning Tree Protocol (VSTP) for a virtual switch routing instance.

**vpls**—Specify VPLS as the protocol for a routing instance.

**Required Privilege Level** **routing**—To view this statement in the configuration.  
**routing-control**—To add this statement to the configuration.

**Related Documentation**

- *Example: Configuring Multiple Routing Instances of OSPF*

## receive (Protocols BGP)

---

<b>Syntax</b>	receive;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family <i>family</i> add-path], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor address family <i>family</i> add-path], [edit protocols bgp group <i>group-name</i> family <i>family</i> add-path], [edit protocols bgp group <i>group-name</i> family <i>family</i> add-path neighbor address family <i>family</i> add-path]
<b>Release Information</b>	Statement introduced in Junos OS Release 11.3.
<b>Description</b>	Enable the router to receive multiple paths to a destination. You can enable the router to receive multiple paths from specified neighbors or from all neighbors.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Advertising Multiple BGP Paths to a Destination on page 392</a></li></ul>

## record (Origin Validation for BGP)

<b>Syntax</b>	<pre>record <i>destination</i> {     <b>maximum-length</b> <i>prefix-length</i> {         <b>origin-autonomous-system</b> <i>as-number</i> {             <b>validation-state</b> (invalid   valid);         }     } }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation static],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation static],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation static],</p> <p>[edit routing-options validation static]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	<p>Configure the network prefix for the route validation (RV) record.</p> <p>An RV record matches any route whose prefix matches the RV prefix, whose prefix length does not exceed the <b>maximum-length</b> given in the RV record, and whose origin AS equals the <b>origin-autonomous-system</b> given in the RV record. RV records are received from the cache server using the protocol defined in Internet draft draft-ietf-sidr-rpki-rtr-19, <i>The RPKI/Router Protocol</i>, and can also be configured statically, as shown here.</p>
<b>Options</b>	<p><b>destination</b>—Network prefix for the RV record.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## record-lifetime (Origin Validation for BGP)


---

<b>Syntax</b>	<code>record-lifetime <i>seconds</i>;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i> ], [edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i> ], [edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i> ], [edit routing-options validation group <i>group-name</i> session <i>server-ip-address</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	Configure the amount of time that route validation (RV) records learned from a cache server are valid. RV records expire if the session to the cache server goes down and remains down for the <b>record-lifetime</b> (seconds).
<b>Options</b>	<b><i>seconds</i></b> —Amount of time that an RV remains valid after the session to the cache server goes down. <b>Range:</b> 60 (one minute) through 604800 (one week) <b>Default:</b> 3600 seconds (one hour)
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## refresh-time (Origin Validation for BGP)

<b>Syntax</b>	<code>refresh-time seconds;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation group <i>group-name</i> session <i>server-ip-address</i>],</p> <p>[edit routing-options validation group <i>group-name</i> session <i>server-ip-address</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2.
<b>Description</b>	<p>Configure a liveliness check interval for a configured resource public key infrastructure (RPKI) cache server. Every <b>refresh-time</b> (seconds), a serial query protocol data unit (PDU) with the last known serial number is transmitted. The <b>refresh-time</b> cannot be longer than half of the <a href="#">hold-time</a>.</p>
<b>Options</b>	<p><b>seconds</b>—Interval at which serial query PDUs are sent.</p> <p><b>Range:</b> 1 through 1800</p> <p><b>Default:</b> 300</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## remove-private

<b>Syntax</b>	remove-private all replace nearest;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>neighbor</b> <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>When advertising AS paths to remote systems, have the local system strip private AS numbers from the AS path. The numbers are stripped from the AS path starting at the left end of the AS path (the end where AS paths have been most recently added). The routing device stops searching for private ASs when it finds the first nonprivate AS or a peer's private AS. If the AS path contains the AS number of the external BGP (EBGP) neighbor, BGP does not remove the private AS number.</p>
<div>  <p><b>NOTE:</b> As of Junos OS 10.0R2 and higher, if there is a need to send prefixes to an EBGP peer that has an AS number that matches an AS number in the AS path, consider using the <code>as-override</code> statement instead of the <code>remove-private</code> statement.</p> </div>	
<p>The operation takes place after any confederation member ASs have already been removed from the AS path, if applicable.</p> <p>The Junos OS recognizes the set of AS numbers that is considered private, a range that is defined in the Internet Assigned Numbers Authority (IANA) assigned numbers document.</p> <p>The set of reserved AS numbers is in the range from 64,512 through 65,535.</p>	
<b>Options</b>	<p><b>all</b>—Remove all private AS numbers from the original path. Do not stop the process of removing private AS numbers, even if a public AS number is encountered.</p>

**nearest**—When you use the **all** and **replace** options, choose the last (right-most) public AS number encountered in the original AS path for the replacement value, as the AS path is processed from left to right. If no public AS number is encountered, the default replacement value is used. (See the **replace** option for information about the default replacement value.)

**replace**—When you use the **all** option, instead of removing private AS numbers, perform a replace operation. The default replacement value for the private AS number is the local AS number at the BGP group level for the BGP peer. If you are unsure about the replacement value, check the local AS value displayed in the output of the **show bgp group group-name** command.

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Removing Private AS Numbers from AS Paths on page 269</a></li></ul>
------------------------------	--

## resolve-vpn

<b>Syntax</b>	resolve-vpn;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>family</b> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet labeled-unicast],</p> <p>[edit protocols bgp <b>family</b> inet labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> <b>family</b> inet labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet labeled-unicast]</p>
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	Allow labeled routes to be placed in the inet.3 routing table for route resolution. These routes are then resolved for PE router connections where the remote PE is located across another AS. For a PE router to install a route in the VRF, the next hop must resolve to a route stored within the inet.3 table.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Multiprotocol BGP on page 537</a></li> </ul>

## restart-time (BGP Graceful Restart)

<b>Syntax</b>	<code>restart-time seconds;</code>
<b>Hierarchy Level</b>	<p>[edit protocols (bgp   rip   ripng) graceful-restart],          [edit logical-systems <i>logical-system-name</i> protocols (bgp   rip   ripng) graceful-restart (Enabling Globally)],          [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp graceful-restart],          [edit routing-instances <i>routing-instance-name</i> protocols bgp graceful-restart]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.3.          Statement introduced in Junos OS Release 9.0 for EX Series switches.          Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	Configure the duration of the BGP, RIP, or next-generation RIP (RIPng) graceful restart period.
<b>Options</b>	<p><b>seconds</b>—Length of time for the graceful restart period.  <b>Range:</b> 1 through 600 seconds  <b>Default:</b> Varies by protocol:</p> <ul style="list-style-type: none"> <li>• BGP—120 seconds</li> <li>• RIP and RIPng—60 seconds</li> </ul>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.          routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring Graceful Restart Options for BGP</i></li> <li>• <i>Configuring Graceful Restart Options for RIP and RIPng</i></li> <li>• <i>Configuring Graceful Restart for QFabric Systems</i></li> <li>• <a href="#">stale-routes-time on page 762</a></li> </ul>

## rib (Protocols BGP)

<b>Syntax</b>	<code>rib (inet.3   inet6.3) ;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit protocols bgp <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> (inet   inet6) labeled-unicast],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> (inet   inet6) labeled-unicast]</p>
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	<p>You can allow both labeled and unlabeled routes to be exchanged in a single session. The labeled routes are placed in the inet.3 or inet6.3 routing table, and both labeled and unlabeled unicast routes can be sent or received by the router.</p>
<b>Options</b>	<p><b>inet.3</b>—Name of the routing table for IPv4.</p> <p><b>inet6.3</b>—Name of the routing table for IPv6.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Examples: Configuring Multiprotocol BGP on page 537</a></li> </ul>

## rib-group (Protocols BGP)

<b>Syntax</b>	<code>rib-group group-name;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit protocols bgp <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit protocols bgp group <i>group-name</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> <b>family</b> inet (labeled-unicast   unicast   multicast)],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> <b>family</b> inet (labeled-unicast   unicast   multicast)]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Add unicast prefixes to unicast and multicast tables.
<b>Options</b>	<b>group-name</b> —Name of the routing table group. The name must start with a letter and can include letters, numbers, and hyphens. You generally specify only one routing table group.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Exporting Specific Routes from One Routing Table Into Another Routing Table</i></li> <li>• <i>Example: Importing Direct and Static Routes Into a Routing Instance</i></li> <li>• <a href="#">Understanding Multiprotocol BGP on page 537</a></li> </ul>

## route-monitoring

<b>Syntax</b>	<pre> route-monitoring {     none;     post-policy {         exclude-non-eligible;     }     pre-policy {         exclude-non-feasible;     } } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>bmp</b>],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bmp],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bmp],  [edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>],  [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station</b> <i>station-name</i>],  [edit protocols bgp <b>bmp</b>],  [edit protocols bgp group <i>group-name</i> bmp],  [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bmp],  [edit routing-options <b>bmp</b>],  [edit routing-options bmp <b>station</b> <i>station-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series.  Statement introduced in Junos OS Release 13.3.</p>
<b>Description</b>	<p>Specify whether BMP should send pre-policy route monitoring messages, post-policy route monitoring messages, both types of messages, or none at all. The pre-policy can be configured to exclude routes that are non-feasible for the decision process (for example, a route loop). The post-policy can be configured to exclude routes that are not eligible for the decision process (for example, protocol nexthop not resolved).</p> <p>You can also selectively enable or disable BMP route monitoring at various hierarchy levels (for example, [edit protocols bgp group <i>group-name</i>] or [edit protocols bgp group <i>group-name</i> neighbor <i>address</i>]).</p>
<b>Options</b>	<p><b>none</b>—Explicitly disables BMP route monitoring.</p> <p><b>Default:</b> If you configure the <b>route-monitoring</b> statement at the [edit routing-options <b>bmp</b>] hierarchy level, the default option is <b>pre-policy</b>. If you configure the <b>route-monitoring</b> statement at any of the [edit protocols bgp] hierarchy levels, the default option is to inherit the configuration from the <b>route-monitoring</b> statement configured at the [edit routing-options <b>bmp</b>] hierarchy level.</p> <p>The other statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.  routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## route-target (Protocols BGP)

<b>Syntax</b>	<pre> route-target {     accepted-prefix-limit {         maximum <i>number</i>;         teardown &lt;<i>percentage</i>&gt; &lt;idle-timeout (forever   <i>time-in-minutes</i>)&gt;;     }     advertise-default;     external-paths <i>number</i>;     prefix-limit {         maximum <i>number</i>;         teardown &lt;<i>percentage</i>&gt; &lt;idle-timeout (forever   <i>time-in-minutes</i>)&gt;;     } } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name neighbor address family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name family</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address family</i>],</p> <p>[edit protocols bgp <i>family</i>],</p> <p>[edit protocols bgp <i>group group-name family</i>],</p> <p>[edit protocols bgp <i>group group-name neighbor address family</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name family</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name neighbor address family</i>]</p>
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	Limit the number of prefixes advertised on BGP peers specifically to the peers that need the updates.
<b>Options</b>	<p><b>advertise-default</b>—Advertise default routes and suppress more specific routes.</p> <p><b>external-paths <i>number</i></b>—Number of external paths accepted for route filtering.</p> <p><b>Range:</b> 1 through 256 paths</p> <p><b>Default:</b> 1 path</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Configuring an Export Policy for BGP Route Target Filtering</i></li> <li>• <i>Example: Configuring Proxy BGP Route Target Filtering</i></li> </ul>

## routing-instances (Multiple Routing Entities)

---

<b>Syntax</b>	<code>routing-instances <i>routing-instance-name</i> { ... }</code>
<b>Hierarchy Level</b>	[edit], [edit logical-systems <i>logical-system-name</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	<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"><li>• A set of routing tables</li><li>• A set of interfaces that belong to these routing tables</li><li>• A set of routing option configurations</li></ul> <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 <b>my-instance</b>, its corresponding IP unicast table is my-instance.inet.0. All routes for <b>my-instance</b> 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>
<b>Default</b>	Routing instances are disabled for the router.
<b>Options</b>	<p><b><i>routing-instance-name</i></b>—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>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Example: Configuring Interprovider Layer 3 VPN Option A](#)
  - [Example: Configuring Interprovider Layer 3 VPN Option B](#)
  - [Example: Configuring Interprovider Layer 3 VPN Option C](#)

## send (Logical Systems Add-Path)

<b>Syntax</b>	<pre>send {   path-count number;   prefix-policy [ policy-names ]; }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family inet unicast add-path],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor address family inet unicast add-path],</p> <p>[edit protocols bgp group <i>group-name</i> family inet unicast add-path],</p> <p>[edit protocols bgp group <i>group-name</i> family inet unicast add-path neighbor address family inet unicast add-path]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 11.3.
<b>Description</b>	<p>Enable advertisement of multiple paths to a destination, instead of advertising only the active path.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Advertising Multiple BGP Paths to a Destination on page 392</a></li> </ul>

## session (Origin Validation for BGP)

---

<b>Syntax</b>	<pre>session address {     hold-time seconds;     local-address local-ip-address;     port port-number;     preference number;     record-lifetime seconds;     refresh-time seconds; }</pre>
<b>Hierarchy Level</b>	<pre>[edit logical-systems logical-system-name routing-instances instance-name routing-options validation group group-name], [edit logical-systems logical-system-name routing-options validation group group-name], [edit routing-instances instance-name routing-options validation group group-name], [edit routing-options validation group group-name]</pre>
<b>Release Information</b>	Statement introduced in Junos OS Release 12.2. Statement introduced in Junos OS Release 12.3 for ACX Series routers.
<b>Description</b>	<p>Configure a secure shell (SSH) session with a resource public key infrastructure (RPKI) cache server. The router-to-cache transport protocol is carried using a TCP session to a configurable port. Caches are organized in groups. The Junos OS implementation supports up to 63 sessions per group and both IPv4 and IPv6 address families.</p> <p>The maximum number of sessions in a group is two, by default, and is configurable. If the number of sessions in a group exceeds the <a href="#">max-sessions</a> value, the connections are established in order by <a href="#">preference</a> value. A numerically higher preference results in a higher probability for session establishment. The order of session establishment is random among sessions with equal preferences.</p>
<b>Options</b>	<p><b>address</b>—IP address of the cache server.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>

## session-mode

<b>Syntax</b>	session-mode (automatic   multihop   single-hop);
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 11.1.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p>
<b>Description</b>	<p>Configure BFD session mode to be single-hop or multihop. By default, BGP uses single-hop BFD sessions if the peer is directly connected to the router's interface. BGP uses multihop BFD sessions if the peer is not directly connected to the router's interface. If the peer session's <b>local-address</b> option is configured, the directly connected check is based partly on the source address that would be used for BGP and BFD.</p> <p>For backward compatibility, you can override the default behavior by configuring the <b>single-hop</b> or <b>multihop</b> option. Before Junos OS Release 11.1, the behavior was to assume that IBGP peer sessions were multihop.</p>
<b>Options</b>	<p><b>automatic</b>—Configure BGP to use single-hop BFD sessions if the peer is directly connected to the router's interface, and multihop BFD sessions if the peer is not directly connected to the router's interface</p> <p><b>multihop</b>—Configure BGP to use multihop BFD sessions.</p> <p><b>single-hop</b>—Configure BGP to use single-hop BFD sessions.</p> <p><b>Default:</b> automatic</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring BFD Authentication for BGP on page 360</a></li> <li>• <a href="#">Example: Configuring BFD on Internal BGP Peer Sessions on page 350</a></li> </ul>

- [Example: Configuring BFD Authentication for BGP on page 360](#)
- [Understanding BFD Authentication for BGP on page 358](#)

---

## stale-routes-time

---

Syntax	stale-routes-time <i>seconds</i> ;
Hierarchy Level	[edit logical-systems <i>logical-routing-name</i> protocols bgp graceful-restart], [edit logical-systems <i>logical-routing-name</i> routing-instances <i>routing-instance-name</i> protocols bgp graceful-restart], [edit protocols bgp graceful-restart], [edit routing-instances <i>routing-instance-name</i> protocols bgp graceful-restart]
Release Information	Statement introduced in Junos OS Release 8.3. 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 maximum time that stale routes are kept during a restart. The <b>stale-routes-time</b> statement allows you to set the length of time the routing device waits to receive messages from restarting neighbors before declaring them down.
Options	<b>seconds</b> —Time the router device waits to receive messages from restarting neighbors before declaring them down. <b>Range:</b> 1 through 600 seconds <b>Default:</b> 300 seconds
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"><li>• <i>Configuring Graceful Restart Options for BGP</i></li><li>• <i>Configuring Graceful Restart for QFabric Systems</i></li><li>• <a href="#">restart-time (BGP Graceful Restart) on page 753</a></li></ul>

## static (Origin Validation for BGP)

<b>Syntax</b>	<pre>static {   record destination {     maximum-length prefix-length {       origin-autonomous-system as-number {         validation-state (invalid   valid);       }     }   } }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation],</p> <p>[edit routing-options validation]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	<p>Configure a static route validation (RV) record.</p> <p>RV records are received from the cache server using the protocol defined in Internet draft draft-ietf-sidr-rpki-rtr-19, <i>The RPKI/Router Protocol</i>, and can also be configured statically, as shown here.</p> <p>Static records are useful for overwriting the information received from an RPKI cache server.</p> <p>An RV record matches any route whose prefix matches the RV prefix <b>record</b>, whose prefix length does not exceed the <b>maximum-length</b> given in the RV record, and whose origin AS equals the <b>origin-autonomous-system</b> number given in the RV record.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## station

<b>Syntax</b>	<pre> station <i>station-name</i> {   authentication-algorithm (aes-128-cmac-96   hmac-sha-1-96   md5);   authentication-key <i>key</i>;   authentication-key-chain <i>authentication-key-chain</i>;   connection-mode (active   passive);   hold-down {     seconds;     flaps <i>flaps</i>;     period <i>seconds</i>;   }   initiation-message <i>text</i>;   local-address <i>address</i>;   local-port <i>port</i>;   monitor (disable   enable);   priority (high   low   medium);   route-monitoring {     none;     post-policy {       exclude-non-eligible;     }     pre-policy {       exclude-non-feasible;     }   }   station-address (<i>ip-address</i>   <i>name</i>);   station-port <i>port-number</i>;   statistics-timeout <i>seconds</i>;   traceoptions {     file <i>filename</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;;     flag <i>flag</i> &lt;flag-modifier&gt;;   } } </pre>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options bmp], [edit routing-options bmp]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Specify and configure a BMP monitoring station. Be aware that each BMP monitoring station can use a significant amount of a device's resources. You can configure up to 3 BMP monitoring stations.
<b>Options</b>	<p><b><i>station-name</i></b>—Specify a name for the BMP station.</p> <p>The other statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

## station-address

---

<b>Syntax</b>	<code>station-address (address   station-name);</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <a href="#">bmp</a> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <a href="#">station</a> <i>station-name</i> ], [edit routing-options <a href="#">bmp</a> ], [edit routing-options bmp <a href="#">station</a> <i>station-name</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Specify the name or address for the BMP monitoring station. You can specify one or the other but not both.
<b>Options</b>	<b><i>station-address</i></b> —Specify the address for the BMP station. The address should be a valid IPv4 or IPv6 address.  <b><i>station-name</i></b> —Specify the name for the BMP station.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## station-port

---

<b>Syntax</b>	<code>station-port port;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <a href="#">bmp</a> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <a href="#">station station-name</a> ], [edit routing-options <a href="#">bmp</a> ], [edit routing-options bmp <a href="#">station station-name</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Specify the port number for the BMP monitoring station.
<b>Options</b>	<b>port</b> —Specify the port number for the BMP monitoring station. If the <a href="#">connection-mode</a> statement is configured as <b>active</b> a station port number is required. If the <b>connection-mode</b> statement is configured as <b>passive</b> , you must not configure a station port number. <b>Range:</b> 1 through 65535
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li><li>• <a href="#">connection-mode on page 637</a></li></ul>

## statistics-timeout

---

<b>Syntax</b>	<code>statistics-timeout <i>seconds</i>;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-options <a href="#">bmp</a> ], [edit logical-systems <i>logical-system-name</i> routing-options bmp <a href="#">station</a> <i>station-name</i> ], [edit routing-options <a href="#">bmp</a> ], [edit routing-options bmp <a href="#">station</a> <i>station-name</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Statement introduced in Junos OS Release 13.3.
<b>Description</b>	Specify how often statistics messages are sent to the BMP monitoring station. If you configure a value of 0, no statistics messages are sent.
<b>Options</b>	<b><i>seconds</i></b> —Specify the number for the BMP monitoring station. <b>Default:</b> 3600 seconds <b>Range:</b> 15 through 65535 seconds
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring BGP Monitoring Protocol Version 3 on page 577</a></li> </ul>

## tcp-aggressive-transmission

---

<b>Syntax</b>	tcp-aggressive-transmission;
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-address</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-address</i>], [edit protocols bgp], [edit protocols bgp group <i>group-name</i>], [edit protocols bgp group <i>group-name</i> neighbor <i>neighbor-address</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-address</i>]</p>
<b>Release Information</b>	Statement introduced in Junos OS Release 13.3 for the T Series.
<b>Description</b>	Enables a TCP socket option for the affected BGP sessions, which prioritizes pure ACKs and does not exponentially back-off retransmission for couple of retransmissions.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">[edit protocols bgp] Hierarchy Level on page 597</a></li><li>• <a href="#">[edit routing-instances] Hierarchy Level</a></li></ul>

## tcp-mss (Protocols BGP)

<b>Syntax</b>	<code>tcp-mss <i>segment-size</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-name</i>],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-name</i>],  [edit protocols bgp],  [edit protocol bgp group <i>group-name</i>],  [edit protocols bgp group <i>group-name</i> neighbor <i>neighbor-name</i>],  [edit routing-instances <i>routing-instance-name</i> protocols bgp],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Configure the maximum segment size (MSS) for the TCP connection for BGP neighbors.</p> <p>The MSS is only valid in increments of 2 KB. The value used is based on the value set, but is rounded down to the nearest multiple of 2048.</p>
<b>Options</b>	<p><b><i>segment-size</i></b>—MSS for the TCP connection.</p> <p><b>Range:</b> 1 through 4096</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Limiting TCP Segment Size for BGP on page 468</a></li> </ul>

## threshold (detection-time)

<b>Syntax</b>	<code>threshold <i>milliseconds</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection detection-time],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection detection-time],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection detection-time],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection detection-time],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection detection-time],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection detection-time],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection detection-time],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection detection-time],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection detection-time],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection detection-time],</p> <p>[edit protocols bgp bfd-liveness-detection detection-time],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection detection-time],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection detection-time],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection detection-time]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.2.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPNs and VPLS.</p>
<b>Description</b>	Specify the 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.



**NOTE:** The threshold value must be equal to or greater than the transmit interval.

The threshold time must be equal to or greater than the value specified in the `minimum-interval` or the `minimum-receive-interval` statement.

**Options** *milliseconds*—Value for the detection time adaptation threshold.  
**Range:** 1 through 255,000

**Required Privilege** routing—To view this statement in the configuration.  
**Level** routing-control—To add this statement to the configuration.

**Related Documentation**

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*

## threshold (transmit-interval)

<b>Syntax</b>	<code>threshold <i>milliseconds</i>;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection transmit-interval],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection transmit-interval]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.2.</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 QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Specify 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.

**Options** *milliseconds*—Value for the transmit interval adaptation threshold.

**Range:** 0 through 4,294,967,295 ( $2^{32} - 1$ )



**NOTE:** The threshold value specified in the `threshold` statement must be greater than the value specified in the `minimum-interval` statement for the `transmit-interval` statement.

**Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**


- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- [bfd-liveness-detection on page 627](#)

## topology (Protocols BGP)

<b>Syntax</b>	<pre> topology <i>name</i> {     community {         target <i>identifier</i>;     } } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp family (inet   inet6) unicast],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family (inet   inet6) unicast],  [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet6) unicast],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp family (inet   inet6) unicast],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> family (inet   inet6) unicast],  [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet6) unicast],  [edit protocols bgp family (inet   inet6) unicast],  [edit protocols bgp group <i>group-name</i> family (inet   inet6) unicast],  [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet6) unicast],  [edit routing-instances <i>routing-instance-name</i> protocols bgp family (inet   inet6) unicast],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> family (inet   inet6) unicast],  [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> family (inet   inet6)]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0.  Statement introduced in Junos OS Release 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Enable a topology for BGP multitopology routing. You must first configure one or more topologies under the <b>[edit routing-options]</b> hierarchy level.</p> <p>Apply the community tags to identify the application topologies by configuring a routing topology name and BGP community value.</p> <p>In Junos OS, multitopology support for BGP is based on the community value in a BGP route. This configuration determines the association between a topology and one or more community values and populates the topology routing tables. Arriving BGP updates that have a matching community value are replicated in the associated topology routing table. You decide which BGP community values are associated with a given topology.</p> <p>For example, you can create a configuration that causes BGP updates that are received with community value <b>target:40:40</b> to be added into topology routing table <b>:voice.inet.0</b> (in addition to the default routing table <b>inet.0</b>). Likewise, you configuration can specify that updates that are received with community value <b>target:50:50</b> are added into topology routing table <b>:video.inet.0</b> (in addition to the default routing table <b>inet.0</b>).</p>
<b>Options</b>	<p><b>name</b>—Name of a topology you configured at the <b>[edit routing-options]</b> hierarchy level to create a topology for a specific type of traffic, such as voice or video.</p>

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Configuring Multitopology Routing Based on Applications</i></li><li>• <i>Example: Configuring Multitopology Routing Based on a Multicast Source</i></li></ul>

## traceoptions (Protocols BGP)

<b>Syntax</b>	<pre> traceoptions {     file <i>filename</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;;     flag <i>flag</i> &lt;flag-modifier&gt; &lt;disable&gt;; } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit protocols bgp],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>4byte-as</b> statement introduced in Junos OS Release 9.2.</p> <p><b>4byte-as</b> statement introduced in Junos OS Release 9.2 for EX Series switches.</p>
<b>Description</b>	Configure BGP protocol-level tracing options. To specify more than one tracing operation, include multiple flag statements.
<div>  <b>NOTE:</b> The <b>traceoptions</b> statement is not supported on QFabric systems. </div>	
<b>Default</b>	<p>The default BGP protocol-level tracing options are inherited from the routing protocols <b>traceoptions</b> statement included at the [edit routing-options] hierarchy level. The default group-level trace options are inherited from the BGP protocol-level <b>traceoptions</b> statement. The default peer-level trace options are inherited from the group-level <b>traceoptions</b> statement.</p>
<b>Options</b>	<p><b>disable</b>—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as <b>all</b>.</p> <p><b>file <i>name</i></b>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory <b>/var/log</b>. We recommend that you place BGP tracing output in the file <b>bgp-log</b>.</p>

**files *number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file.0*** 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**—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

#### BGP Tracing Flags

- **4byte-as**—4-byte AS events.
- **bfd**—BFD protocol events.
- **damping**—Damping operations.
- **graceful-restart**—Graceful restart events.
- **keepalive**—BGP keepalive messages. If you enable the the BGP **update** flag only, received keepalive messages do not generate a trace message.
- **nsr-synchronization**—Nonstop routing synchronization events.
- **open**—Open packets. These packets are sent between peers when they are establishing a connection.
- **packets**—All BGP protocol packets.
- **refresh**—BGP refresh packets.
- **update**—Update packets. These packets provide routing updates to BGP systems. If you enable only this flag, received keepalive messages do not generate a trace message. Use the **keepalive** flag to generate a trace message for keepalive messages.

#### Global Tracing Flags

- **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.
- **filter**—Provide filter trace information. Applies only to **route**, **damping**, and **update** tracing flags.
- **receive**—Trace the packets being received.
- **send**—Trace the 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), megabytes (MB), or gigabytes (GB). 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 also must 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.

<b>Required Privilege Level</b>	routing and trace—To view this statement in the configuration.
	routing-control and trace-control—To add this statement to the configuration.
<b>Related Documentation</b>	• <a href="#">log-updown on page 689</a> statement
	• <i>Tracing Nonstop Active Routing Synchronization Events</i>
	• <a href="#">Understanding Trace Operations for BGP Protocol Traffic on page 583</a>
	• <i>Configuring OSPF Refresh and Flooding Reduction in Stable Topologies</i>

## traceoptions (Protocols BMP)

<b>Syntax</b>	<pre>traceoptions {     file <i>file-name</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;;     flag <i>flag</i> &lt;<i>flag-modifier</i>&gt; &lt;disable&gt;; }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-options <b>bmp</b>],          [edit logical-systems <i>logical-system-name</i> routing-options bmp <b>station</b> <i>station-name</i>],          [edit routing-options <b>bmp</b>],          [edit routing-options bmp <b>station</b> <i>station-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 13.2X51-D15 for the QFX Series.          Statement introduced in Junos OS Release 13.3.</p>
<b>Description</b>	<p>Configure tracing options for BMP monitoring. To specify more than one tracing operation, include multiple flag statements.</p>
<b>Options</b>	<p><b>file</b> <i>file-name</i>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory <b>/var/log</b>. We recommend that you place BMP tracing output in the file <b>bmp-log</b>.</p> <p><b>files</b> <i>number</i>—(Optional) Maximum number of trace files. When a trace file named <b>trace-file.0</b> reaches its maximum size, it is renamed <b>trace-file.0</b>, then <b>trace-file.1</b>, 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 <b>size</b> option.</p> <p><b>Range:</b> 2 through 1000 files  <b>Default:</b> 10 files</p> <p><b>flag</b>—Tracing operation to perform. To specify more than one tracing operation, include multiple <b>flag</b> statements.</p> <ul style="list-style-type: none"> <li>• <b>all</b>—Trace all BMP monitoring operations.</li> <li>• <b>down</b>—Down messages.</li> <li>• <b>error</b>—Error conditions.</li> <li>• <b>event</b>—Major events, station establishment, errors, and events.</li> <li>• <b>general</b>—General events.</li> <li>• <b>normal</b>—Normal events.</li> <li>• <b>packets</b>—All messages.</li> <li>• <b>policy</b>—Policy processing.</li> <li>• <b>route</b>—Routing information.</li> <li>• <b>route-monitoring</b>—Route monitoring messages.</li> <li>• <b>state</b>—State transitions.</li> </ul>

- **statistics**—Statistics messages.
- **task**—Routing protocol task processing.
- **timer**—Routing protocol timer processing.
- **up**—Up messages.
- **write**—Writing of messages.

**flag-modifier**—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Provide detailed trace information.
- **disable**—Disable the tracing flag.
- **receive**—Trace the packets being received.
- **send**—Trace the 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), megabytes (MB), or gigabytes (GB). 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 also must 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.

<b>Required Privilege Level</b>	routing and trace—To view this statement in the configuration. routing-control and trace-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Tracing BMP Operations on page 589</a></li><li>• <a href="#">Understanding Trace Operations for BGP Protocol Traffic on page 583</a></li><li>• <a href="#">Configuring OSPF Refresh and Flooding Reduction in Stable Topologies</a></li></ul>

## traceoptions (Origin Validation for BGP)

---

<b>Syntax</b>	<pre>traceoptions {   file <i>filename</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;;   flag <i>flag</i>; }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation],</p> <p>[edit routing-options validation]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	<p>Configure tracing operations for resource public key infrastructure (RPKI) BGP route validation.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## traffic-statistics (Protocols BGP)

---

<b>Syntax</b>	<pre>traffic-statistics {     file <i>filename</i> &lt;world-readable   no-world-readable&gt;;     interval <i>seconds</i>; }</pre>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp family (inet   inet6) labeled-unicast], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> family (inet   inet6) labeled-unicast], [edit protocols bgp family (inet   inet6) labeled-unicast], [edit protocols bgp group <i>group-name</i> family (inet   inet6) labeled-unicast]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	Enable the collection of traffic statistics for interprovider or carrier-of-carriers VPNs.
<b>Options</b>	<p><b>file <i>filename</i></b>—Specify a filename for the BGP labeled-unicast traffic statistics file. If you do not specify a filename, statistics are still collected but can only be viewed by using the <b>show bgp group traffic statistics <i>group-name</i></b> command.</p> <p><b>interval <i>seconds</i></b>—Specify how often BGP labeled-unicast traffic statistics are collected.</p>
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics</i></li></ul>

## transmit-interval (BFD Liveness Detection)

<b>Syntax</b>	<pre>transmit-interval {     minimum-interval milliseconds;     threshold milliseconds; }</pre>
<b>Hierarchy Level</b>	<pre>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i>   bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> bfd-liveness-detection], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   l2vpn oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   vpls oam bfd-liveness-detection], [edit protocols bgp bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> bfd-liveness-detection], [edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i>   bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor   <i>address</i> bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam   bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i>   neighbor <i>neighbor-id</i> oam bfd-liveness-detection], [edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</pre>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.2.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for BFD authentication introduced in Junos OS Release 9.6.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	<p>Specify the transmit interval for the <b>bfd-liveness-detection</b> statement. The negotiated transmit interval for a peer is the interval between the sending of BFD packets to peers. The receive interval for a peer is the minimum time that it requires between packets sent from its peer; the receive interval is not negotiated between peers. To determine the transmit interval, each peer compares its configured minimum transmit interval with its</p>

peer's minimum receive interval. The larger of the two numbers is accepted as the transmit interval for that peer.

The remaining statements are explained separately.

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Configuring BFD for Layer 2 VPN and VPLS</i></li><li>• <i>Example: Configuring BFD for Static Routes</i></li><li>• <a href="#">bfd-liveness-detection on page 627</a></li><li>• <a href="#">threshold on page 772</a></li><li>• <a href="#">minimum-interval on page 702</a></li><li>• <a href="#">minimum-receive-interval on page 704</a></li></ul>
------------------------------	--

## ttl (Protocols BGP)

<b>Syntax</b>	<code>ttl <i>ttl-value</i>;</code>
<b>Hierarchy Level</b>	<pre> [edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp multihop], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i> multihop], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name</i>   <i>neighbor address</i>], [edit logical-systems <i>logical-system-name</i> protocols bgp <i>group group-name neighbor address</i>   multihop], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp multihop], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp <i>group group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp <i>group group-name</i> multihop], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp <i>group group-name neighbor address</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols   bgp <i>group group-name neighbor address</i> multihop], [edit protocols bgp], [edit protocols bgp multihop], [edit protocols bgp <i>group group-name</i> multihop], [edit protocols bgp <i>group group-name neighbor address</i>], [edit protocols bgp <i>group group-name neighbor address</i> multihop], [edit routing-instances <i>routing-instance-name</i> protocols bgp], [edit routing-instances <i>routing-instance-name</i> protocols bgp multihop], [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i> multihop], [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>   <i>neighbor address</i>] [edit routing-instances <i>routing-instance-name</i> protocols bgp <i>group group-name</i>   <i>neighbor address</i> multihop] </pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p>Support for setting the TTL on single-hop external BGP (EBGP) peers introduced in Junos OS Release 13.3.</p>
<b>Description</b>	<p>Configure the maximum time-to-live (TTL) value for the TTL in the IP header of BGP packets.</p> <p>For BGP multihop scenarios, in which EBGP peers are not directly connected to each other, setting a TTL is optional. The default setting is 64.</p> <p>For BGP single-hop scenarios, in which external EBGP peers are directly connected to each other, you can, optionally, set the TTL to 255 and configure an inbound firewall filter</p>

to allow only BGP control packets with the TTL set to 255. This is in accordance with RFC 3682, *The Generalized TTL Security Mechanism (GTSM)*. For example:

Send all BGP control packets with the TTL set to 255:

```
user@host# show protocols
bgp {
  group toAS2 {
    type external;
    peer-as 2;
    ttl 255;
    neighbor 10.1.2.3;
    neighbor 10.3.4.5;
    neighbor 10.5.6.7;
  }
}
```

Accept only BGP control packets that have the TTL set to 255:

```
user@host# show firewall
filter ttl-security {
  term gtsm {
    from {
      source-address {
        10.1.2.3/32;
        10.3.4.5/32;
        10.5.6.7/32;
      }
      protocol tcp;
      ttl-except 255;
      port 179;
    }
    then {
      discard;
    }
  }
  term else {
    then {
      accept;
    }
  }
}
```

Apply the firewall filter to the inbound interface for the EBGP single-hop peer:

```
user@host# show interfaces
ge-1/0/0 {
  unit 0 {
    family inet {
      filter {
        input gtsm;
      }
    }
  }
}
```

<b>Options</b>	<p><b><i>ttl-value</i></b>—TTL value for BGP packets.</p> <p><b>Range:</b> 1 through 255, for multihop peers</p> <p><b>Default:</b> 64 (for multihop EBGP sessions, confederations, and IBGP sessions)</p> <p><b>Range:</b> 1 or 255, for single-hop peers</p> <p><b>Default:</b> 1 (for single-hop EBGP sessions)</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring EBGP Multihop Sessions on page 241</a></li> <li>• <a href="#">multihop on page 708</a></li> <li>• <a href="#">no-nexthop-change on page 721</a></li> <li>• <a href="http://www.juniper.net/us/en/community/junos/script-automation/library/configuration/ttl-security/">http://www.juniper.net/us/en/community/junos/script-automation/library/configuration/ttl-security/</a></li> </ul>

## type (Protocols BGP)

<b>Syntax</b>	<code>type type;</code>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit protocols bgp <b>group</b> <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp <b>group</b> <i>group-name</i>]</p>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	<p>Specify the type of BGP peer group.</p> <p>When configuring a BGP group, you can indicate whether the group is an IBGP group or an EBGP group. All peers in an IBGP group are in the same AS, while peers in an EBGP group are in different ASs and normally share a subnet.</p>
<b>Options</b>	<p><b><i>type</i></b>—Type of group:</p> <ul style="list-style-type: none"> <li>• <b>external</b>—External group, which allows inter-AS BGP routing</li> <li>• <b>internal</b>—Internal group, which allows intra-AS BGP routing</li> </ul>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">BGP Feature Guide for Routing Devices</a></li> </ul>

## validation (Origin Validation for BGP)

<b>Syntax</b>	<pre> validation {     group group-name {         max-sessions number;         session address {             hold-time seconds;             local-address local-ip-address;             port port-number;             preference number;             record-lifetime seconds;             refresh-time seconds;         }     }     static {         record destination {             maximum-length prefix-length {                 origin-autonomous-system as-number {                     validation-state (invalid   valid);                 }             }         }     }     traceoptions {         file filename &lt;files number&gt; &lt;size size&gt; &lt;world-readable   no-world-readable&gt;;         flag flag;     } } </pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options],  [edit logical-systems <i>logical-system-name</i> routing-options],  [edit routing-instances <i>instance-name</i> routing-options],  [edit routing-options]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.  Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	<p>Configure resource public key infrastructure (RPKI) BGP route validation.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.  routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## validation-state (Origin Validation for BGP)

<b>Syntax</b>	validation-state (invalid   valid);
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> origin-autonomous-system <i>as-number</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> origin-autonomous-system <i>as-number</i>],</p> <p>[edit routing-instances <i>instance-name</i> routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> origin-autonomous-system <i>as-number</i>],</p> <p>[edit routing-options validation static record <i>destination</i> maximum-length <i>prefix-length</i> origin-autonomous-system <i>as-number</i>]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 12.2.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p>
<b>Description</b>	Configure the validation state for a route validation record.
<b>Options</b>	<p><b>invalid</b>—A negative (invalid) validation state. 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.</p> <p><b>valid</b>—A positive (valid) validation state. Indicates that the prefix and AS pair are found in the database.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>

## version (BFD Liveness Detection)

<b>Syntax</b>	version (0   1   automatic);
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit logical-system <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection],</p> <p>[edit protocols bgp bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit protocols bgp group <i>group-name</i> neighbor <i>address</i> bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols bgp group <i>group-name</i> neighbor <i>address</i> bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols l2vpn oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor <i>neighbor-id</i> oam bfd-liveness-detection],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls oam bfd-liveness-detection]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.1</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.</p>
<b>Description</b>	Specify the BFD version for detection. You can explicitly configure BFD version 0, version 1, or the routing device can automatically detect the BFD version. By default, the routing device automatically detects the BFD version, which is either 0 or 1.
<b>Options</b>	<p>Configure the BFD version to detect: <b>0</b> (BFD version 0), <b>1</b> (BFD version 1), or <b>automatic</b> (autodetect the BFD version)</p> <p><b>Default:</b> automatic</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

- Related Documentation**
- [Configuring BFD for Layer 2 VPN and VPLS](#)
  - [Example: Configuring BFD Authentication for BGP on page 360](#)
  - [Example: Configuring BFD on Internal BGP Peer Sessions on page 350](#)
  - [Example: Configuring BFD Authentication for BGP on page 360](#)
  - [Understanding BFD Authentication for BGP on page 358](#)

## vpn-apply-export

<b>Syntax</b>	vpn-apply-export;
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> protocols bgp], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> ], [edit logical-systems <i>logical-system-name</i> protocols bgp group <i>group-name</i> neighbor <i>neighbor</i> ], [edit protocols bgp], [edit protocols bgp group <i>group-name</i> ], [edit protocols bgp group <i>group-name</i> neighbor <i>neighbor</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4.
<b>Description</b>	Apply both the VRF export and BGP group or neighbor export policies (VRF first, then BGP) before routes from the <b>vrf</b> or <b>l2vpn</b> routing tables are advertised to other PE routers.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring Policies for the VRF Table on PE Routers in VPNs</a></li> </ul>



## PART 3

# Administration

- [BGP Operational Commands on page 795](#)



## CHAPTER 16

# BGP Operational Commands

- clear bfd adaptation
- clear bfd session
- clear bgp damping
- clear bgp neighbor
- clear bgp table
- clear validation database
- clear validation session
- clear validation statistics
- monitor traffic
- request validation policy
- restart
- show bfd session
- show bgp bmp
- show bgp group
- show bgp group traffic-statistics
- show bgp neighbor
- show bgp replication
- show bgp replication logical-system
- show bgp summary
- show policy damping
- show policy
- show policy conditions
- show policy damping
- show route
- show route active-path
- show route advertising-protocol
- show route all
- show route aspath-regex

- `show route best`
- `show route brief`
- `show route community`
- `show route community-name`
- `show route damping`
- `show route detail`
- `show route exact`
- `show route export`
- `show route extensive`
- `show route flow validation`
- `show route forwarding-table`
- `show route hidden`
- `show route inactive-path`
- `show route inactive-prefix`
- `show route instance`
- `show route next-hop`
- `show route no-community`
- `show route output`
- `show route protocol`
- `show route receive-protocol`
- `show route table`
- `show route terse`
- `show security keychain`
- `show validation database`
- `show validation group`
- `show validation replication database`
- `show validation session`
- `show validation statistics`
- `test policy`

## clear bfd adaptation

---

<b>Syntax</b>	clear bfd adaptation <address <i>session-address</i> > <discriminator <i>discr-number</i> >
<b>Release Information</b>	Command introduced before Junos OS Release 7.4.
<b>Description</b>	<p>Clear adaptation for Bidirectional Forwarding Detection (BFD) sessions. BFD is a simple hello mechanism that detects failures in a network. Configured BFD interval timers can change, adapting to network situations. Use this command to return BFD interval timers to their configured values.</p> <p>The <b>clear bfd adaptation</b> command is hitless, meaning that the command does not affect traffic flow on the routing device.</p>
<b>Options</b>	<p><b>none</b>—Clear adaptation for all BFD sessions.</p> <p><b>address <i>session-address</i></b>—(Optional) Clear adaptation for all BFD sessions matching the specified address.</p> <p><b>discriminator <i>discr-number</i></b>—(Optional) Clear adaptation for the local BFD session matching the specified discriminator.</p>
<b>Additional Information</b>	For more information, see the description of the <b>bfd-liveness-detection</b> configuration statement in the <i>Junos Routing Protocols Configuration Guide</i> .
<b>Required Privilege Level</b>	clear
<b>List of Sample Output</b>	<a href="#">clear bfd adaptation on page 797</a>
<b>Output Fields</b>	When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### clear bfd adaptation

```
user@host> clear bfd adaptation
```

## clear bfd session

---

<b>List of Syntax</b>	<a href="#">Syntax on page 798</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 798</a>
<b>Syntax</b>	<code>clear bfd session</code> <code>&lt;address <i>session-address</i>&gt;</code> <code>&lt;discriminator <i>discr-number</i>&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
<b>Syntax (EX Series Switch and QFX Series)</b>	<code>clear bfd session</code> <code>&lt;address <i>session-address</i>&gt;</code> <code>&lt;discriminator <i>discr-number</i>&gt;</code>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 12.1 for the QFX Series.
<b>Description</b>	Drop one or more Bidirectional Forwarding Detection (BFD) sessions.
<b>Options</b>	<b>none</b> —Drop all BFD sessions.  <b>address <i>session-address</i></b> —(Optional) Drop all BFD sessions matching the specified address.  <b>discriminator <i>discr-number</i></b> —(Optional) Drop the local BFD session matching the specified discriminator.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show bfd session on page 829</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear bfd session on page 798</a>
<b>Output Fields</b>	When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### clear bfd session

```
user@host> clear bfd session
```

## clear bgp damping

<b>List of Syntax</b>	<a href="#">Syntax on page 799</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 799</a>
<b>Syntax</b>	<pre>clear bgp damping &lt;logical-system (all   <i>logical-system-name</i>)&gt; &lt;prefix&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>clear bgp damping &lt;prefix&gt;</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Clear BGP route flap damping information.
<b>Options</b>	<p><b>none</b>—Clear all BGP route flap damping information.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>prefix</b>—(Optional) Clear route flap damping information for only the specified destination prefix.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show policy damping on page 873</a></li> <li>• <a href="#">show route damping on page 910</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">clear bgp damping on page 799</a>
<b>Output Fields</b>	When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### clear bgp damping

```
user@host> clear bgp damping
```

## clear bgp neighbor

---

<b>List of Syntax</b>	<a href="#">Syntax on page 800</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 800</a>
<b>Syntax</b>	<pre>clear bgp neighbor &lt;as <i>as-number</i>&gt; &lt;instance <i>instance-name</i>&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt; &lt;malformed-route&gt; &lt;neighbor&gt; &lt;soft   soft-inbound&gt; &lt;soft-minimum-igp&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>clear bgp neighbor &lt;as <i>as-number</i>&gt; &lt;instance <i>instance-name</i>&gt; &lt;malformed-route&gt; &lt;neighbor&gt; &lt;soft   soft-inbound&gt; &lt;soft-minimum-igp&gt;</pre>
<b>Release Information</b>	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. <b>malformed-route</b> option introduced in Junos OS Release 13.2.
<b>Description</b>	Perform one of the following tasks: <ul style="list-style-type: none"><li>• Change the state of one or more BGP neighbors to <b>IDLE</b>. For neighbors in the <b>ESTABLISHED</b> state, this command drops the TCP connection to the neighbors and then reestablishes the connection.</li><li>• (<b>soft</b> keyword only) Reapply export policies or import policies, respectively, to one or more BGP neighbors without changing their state.</li><li>• (<b>soft-inbound</b> keyword only) Reapply export policies or import policies, respectively, and send refresh updates to one or more BGP neighbors without changing their state.</li></ul>
<b>Options</b>	<p><b>none</b>—Change the state of all BGP neighbors to <b>IDLE</b>.</p> <p><b>as <i>as-number</i></b>—(Optional) Apply this command only to neighbors in the specified autonomous system (AS).</p> <p><b>instance <i>instance-name</i></b>—(Optional) Apply this command only to neighbors for the specified routing instance.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>malformed-route</b>—(Optional) Remove malformed routes. If a specific neighbor is provided, Junos OS removes malformed routes for that particular neighbor. Otherwise, Junos OS removes malformed routes for all BGP neighbors. To find routes that have</p>

malformed attributes, run the **show route hidden** command, and look for routes marked with **MalformedAttr** in the AS path field.

**neighbor**—(Optional) IP address of a BGP peer. Apply this command only to the specified neighbor.

**soft**—(Optional) Reapply any export policies and send refresh updates to neighbors without clearing the state.

**soft-inbound**—(Optional) Reapply any import policies and send refresh updates to neighbors without clearing the state.

**soft-minimum-igp**—(Optional) Provides soft refresh of the outbound state when the interior gateway protocol (IGP) metric is reset.

**Required Privilege Level**

clear

**Related Documentation**

- [show bgp neighbor on page 848](#)

**List of Sample Output**

[clear bgp neighbor on page 801](#)

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

[clear bgp neighbor](#)

```
user@host> clear bgp neighbor
```

## clear bgp table

---

<b>Syntax</b>	<b>clear bgp table <i>table-name</i></b> <b>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</b>
<b>Syntax (EX Series Switch and QFX Series)</b>	<b>clear bgp table <i>table-name</i></b>
<b>Release Information</b>	Command introduced in Junos OS Release 9.0. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 11.3 for the QFX Series.
<b>Description</b>	Request that BGP refresh routes in a specified routing table.
<b>Options</b>	<b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.  <b><i>table-name</i></b> —Request that BGP refresh routes in the specified table.
<b>Additional Information</b>	In some cases, a prefix limit is associated with a routing table for a VPN instance. When this limit is exceeded (for example, because of a network misconfiguration), some routes might not be inserted in the table. Such routes need to be added to the table after the network issue is resolved. Use the <b>clear bgp table</b> command to request that BGP refresh routes in a VPN instance table.
<b>Required Privilege Level</b>	clear
<b>List of Sample Output</b>	<a href="#">clear bgp table private.inet.0 on page 802</a> <a href="#">clear bgp table inet.6 logical-system all on page 802</a> <a href="#">clear bgp table private.inet.6 logical-system ls1 on page 802</a> <a href="#">clear bgp table logical-system all inet.0 on page 802</a> <a href="#">clear bgp table logical-system ls2 private.inet.0 on page 803</a>
<b>Output Fields</b>	This command produces no output.

## Sample Output

[clear bgp table private.inet.0](#)

```
user@host> clear bgp table private.inet.0
```

[clear bgp table inet.6 logical-system all](#)

```
user@host> clear bgp table inet.6 logical-system all
```

[clear bgp table private.inet.6 logical-system ls1](#)

```
user@host> clear bgp table private.inet.6 logical-system ls1
```

[clear bgp table logical-system all inet.0](#)

```
user@host> clear bgp table logical-system all inet.0
```

`clear bgp table logical-system ls2 private.inet.0`

`user@host> clear bgp table logical-system ls2 private.inet.0`

## clear validation database

---

<b>Syntax</b>	clear validation database <instance <i>instance-name</i> > <logical-system <i>logical-system-name</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Clear the route validation database.
<b>Options</b>	<b>none</b> —Clear the route validation database for all routing instances.  <b>instance <i>instance-name</i></b> —(Optional) Clear the route validation database for the specified instance.  <b>logical-system <i>logical-system-name</i></b> —(Optional) Perform this operation on a particular logical system.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear validation database on page 804</a>

### Sample Output

#### clear validation database

```
user@host> clear validation database
Clearing database
```

## clear validation session

<b>Syntax</b>	clear validation session <destination <i>session-ip-address</i> > <instance <i>instance-name</i> > <logical-system <i>logical-system-name</i> > <soft-inbound>
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Clear the route validation session to the cache server.
<b>Options</b>	<p><b>none</b>—Clear all route validation sessions for all routing instances.</p> <p><b>destination <i>session-ip-address</i></b>—(Optional) Clear the specified route validation session.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Clear the route validation session for the specified instance.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p> <p><b>soft-inbound</b>—(Optional) Rather than flapping the session to the cache server and removing its contents from the database, refresh the session information without removing the database entries.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">clear validation session on page 805</a>

## Sample Output

### clear validation session

```
user@host> clear validation session
Cleared 3 sessions
```

## clear validation statistics

---

<b>Syntax</b>	<code>clear validation statistics</code> <code>&lt;instance <i>instance-name</i>&gt;</code> <code>&lt;logical-system <i>logical-system-name</i>&gt;</code>
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Clear the route validation statistics.
<b>Options</b>	<b>none</b> —Clear the route validation statistics for all routing instances.  <b>instance <i>instance-name</i></b> —(Optional) Clear the route validation statistics for the specified instance.  <b>logical-system <i>logical-system-name</i></b> —(Optional) Perform this operation on a particular logical system.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear validation statistics on page 806</a>

## Sample Output

### clear validation statistics

```
user@host> clear validation statistics
Statistics cleared
```

## monitor traffic

**Syntax** monitor traffic  
 <brief | detail | extensive>  
 <absolute-sequence>  
 <count *count*>  
 <interface *interface-name*>  
 <layer2-headers>  
 <matching *matching*>  
 <no-domain-names>  
 <no-promiscuous>  
 <no-resolve>  
 <no-timestamp>  
 <print-ascii>  
 <print-hex>  
 <resolve-timeout>  
 <size *size*>

**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 11.1 for the QFX Series.

**Description** Display packet headers or packets received and sent from the Routing Engine.



### NOTE:

- Using the **monitor-traffic** command can degrade router or switch performance.
- Delays from DNS resolution can be eliminated by using the **no-resolve** option.



**NOTE:** This command is not supported on the QFabric system.

**Options** **none**—(Optional) Display packet headers transmitted through **fxp0**. On a TX Matrix Plus router, display packet headers transmitted through **em0**.

**brief | detail | extensive**—(Optional) Display the specified level of output.

**absolute-sequence**—(Optional) Display absolute TCP sequence numbers.

**count *count***—(Optional) Specify the number of packet headers to display (0 through 1,000,000). The monitor traffic command quits automatically after displaying the number of packets specified.

**interface *interface-name***—(Optional) Specify the interface on which the **monitor traffic** command displays packet data. If no interface is specified, the **monitor traffic** command displays packet data arriving on the lowest-numbered interface.

**layer2-headers**—(Optional) Display the link-level header on each line.

**matching *matching***—(Optional) Display packet headers that match a regular expression. Use matching expressions to define the level of detail with which the **monitor traffic** command filters and displays packet data.

**no-domain-names**—(Optional) Suppress the display of the domain portion of hostnames. With the **no-domain-names** option enabled, the **monitor traffic** command displays only **team** for the hostname **team.company.net**.

**no-promiscuous**—(Optional) Do not put the interface into promiscuous mode.

**no-resolve**—(Optional) Suppress reverse lookup of the IP addresses.

**no-timestamp**—(Optional) Suppress timestamps on displayed packets.

**print-ascii**—(Optional) Display each packet in ASCII format.

**print-hex**—(Optional) Display each packet, except the link-level header, in hexadecimal format.

**resolve-timeout *timeout***—(Optional) Amount of time the router or switch waits for each reverse lookup before timing out. You can set the timeout for 1 through 4,294,967,295 seconds. The default is 4 seconds. To display each packet, use the **print-ascii**, **print-hex**, or **extensive** option.

**size *size***—(Optional) Read but do not display up to the specified number of bytes for each packet. When set to **brief** output, the default packet size is 96 bytes and is adequate for capturing IP, ICMP, UDP, and TCP packet data. When set to **detail** and **extensive** output, the default packet size is 1514. The **monitor traffic** command truncates displayed packets if the matched data exceeds the configured size.

**Additional Information** In the **monitor traffic** command, you can specify an expression to match by using the **matching** option and including the expression in quotation marks:

**monitor traffic matching "*expression*"**

Replace ***expression*** with one or more of the match conditions listed in [Table 9 on page 809](#).

Table 9: Match Conditions for the monitor traffic Command

Match Type	Condition	Description
Entity	<b>host</b> [ <i>address</i>   <i>hostname</i> ]	Matches packets that contain the specified address or hostname.  The protocol match conditions <b>arp</b> , <b>ip</b> , or <b>rarp</b> , or any of the directional match conditions can be prepended to the <b>host</b> match condition.
	<b>net</b> <i>address</i>	Matches packets with source or destination addresses containing the specified network address.
	<b>net</b> <i>address mask mask</i>	Matches packets containing the specified network address and subnet mask.
	<b>port</b> ( <i>port-number</i>   <i>port-name</i> )	Matches packets containing the specified source or destination TCP or UDP port number or port name.  In place of the numeric port address, you can specify a text synonym, such as <b>bgp</b> (179), <b>dhcp</b> (67), or <b>domain</b> (53) (the port numbers are also listed).
Directional	<b>dst</b>	Matches packets going to the specified destination. This match condition can be prepended to any of the entity type match conditions.
	<b>src</b>	Matches packets from a specified source. This match condition can be prepended to any of the entity type match conditions.
	<b>src and dst</b>	Matches packets that contain the specified source and destination addresses. This match condition can be prepended to any of the entity type match conditions.
	<b>src or dst</b>	Matches packets containing either of the specified addresses. This match condition can be prepended to any of the entity type match conditions.
Packet Length	<b>less</b> <i>value</i>	Matches packets shorter than or equal to the specified value, in bytes.
	<b>greater</b> <i>value</i>	Matches packets longer than or equal to the specified value, in bytes.

Table 9: Match Conditions for the monitor traffic Command (*continued*)

Match Type	Condition	Description
Protocol	<b>amt</b>	Matches all AMT packets. Use the extensive level of output to decode the inner IGMP packets in addition to the AMT outer packet.
	<b>arp</b>	Matches all ARP packets.
	<b>ether</b>	Matches all Ethernet packets.
	<b>ether (broadcast   multicast)</b>	Matches broadcast or multicast Ethernet frames. This match condition can be prepended with <b>src</b> and <b>dst</b> .
	<b>ether protocol (address   (arp   ip   rarp))</b>	Matches packets with the specified Ethernet address or Ethernet packets of the specified protocol type. The <b>ether protocol</b> arguments <b>arp</b> , <b>ip</b> , and <b>rarp</b> are also independent match conditions, so they must be preceded by a backslash (\) when used in the <b>ether protocol</b> match condition.
	<b>icmp</b>	Matches all ICMP packets.
	<b>ip</b>	Matches all IP packets.
	<b>ip (broadcast   multicast)</b>	Matches broadcast or multicast IP packets.
	<b>ip protocol (address   (icmp   igmp   tcp   udp))</b>	Matches packets with the specified address or protocol type. The <b>ip protocol</b> arguments <b>icmp</b> , <b>tcp</b> , and <b>udp</b> are also independent match conditions, so they must be preceded by a backslash (\) when used in the <b>ip protocol</b> match condition.
	<b>isis</b>	Matches all IS-IS routing messages.
	<b>rarp</b>	Matches all RARP packets.
	<b>tcp</b>	Matches all TCP datagrams.
	<b>udp</b>	Matches all UDP datagrams.

To combine expressions, use the logical operators listed in [Table 10 on page 810](#).

Table 10: Logical Operators for the monitor traffic Command

Logical Operator (Highest to Lowest Precedence)	Description
<b>!</b>	Logical NOT. If the first condition does not match, the next condition is evaluated.

Table 10: Logical Operators for the monitor traffic Command (*continued*)

Logical Operator (Highest to Lowest Precedence)	Description
&&	Logical AND. If the first condition matches, the next condition is evaluated. If the first condition does not match, the next condition is skipped.
	Logical OR. If the first condition matches, the next condition is skipped. If the first condition does not match, the next condition is evaluated.
( )	Group operators to override default precedence order. Parentheses are special characters, each of which must be preceded by a backslash (\).

You can use relational operators to compare arithmetic expressions composed of integer constants, binary operators, a length operator, and special packet data accessors. The arithmetic expression matching condition uses the following syntax:

```
monitor traffic matching "ether[0] & 1 != 0"arithmetic_expression relational_operator arithmetic_expression
```

The packet data accessor uses the following syntax:

```
protocol [byte-offset <size>]
```

The optional *size* field represents the number of bytes examined in the packet header. The available values are 1, 2, or 4 bytes. The following sample command captures all multicast traffic:

```
user@host> monitor traffic matching "ether[0] & 1 != 0"
```

To specify match conditions that have a numeric value, use the arithmetic and relational operators listed in [Table 11 on page 812](#).



**NOTE:** Because the Packet Forwarding Engine removes Layer 2 header information before sending packets to the Routing Engine:

- The **monitor traffic** command cannot apply match conditions to inbound traffic.
- The **monitor traffic interface** command also cannot apply match conditions for Layer 3 and Layer 4 packet data, resulting in the match pipe option (`| match`) for this command for Layer 3 and Layer 4 packets not working either. Therefore, ensure that you specify match conditions as described in this command summary. For more information about match conditions, see [Table 9 on page 809](#).
- The 802.1Q VLAN tag information included in the Layer 2 header is removed from all inbound traffic packets. Because the **monitor traffic interface ae[x]** command for aggregated Ethernet interfaces (such as ae0) only shows inbound traffic data, the command does not show VLAN tag information in the output.

**Table 11: Arithmetic and Relational Operators for the monitor traffic Command**

Arithmetic or Relational Operator	Description
<b>Arithmetic Operator</b>	
+	Addition operator.
-	Subtraction operator.
/	Division operator.
&	Bitwise AND.
*	Bitwise exclusive OR.
	Bitwise inclusive OR.
<b>Relational Operator (Highest to Lowest Precedence)</b>	
<=	If the first expression is less than or equal to the second, the packet matches.
>=	If the first expression is greater than or equal to the second, the packet matches.
<	If the first expression is less than the second, the packet matches.
>	If the first expression is greater than the second, the packet matches.
=	If the compared expressions are equal, the packet matches.
!=	If the compared expressions are unequal, the packet matches.

**Required Privilege Level** trace  
maintenance

**List of Sample Output** [monitor traffic count on page 813](#)  
[monitor traffic detail count on page 813](#)  
[monitor traffic extensive \(Absolute Sequence\) on page 813](#)  
[monitor traffic extensive \(Relative Sequence\) on page 813](#)  
[monitor traffic extensive count on page 813](#)  
[monitor traffic interface on page 814](#)  
[monitor traffic matching on page 814](#)  
[monitor traffic \(TX Matrix Plus Router\) on page 814](#)  
[monitor traffic \(QFX3500 Switch\) on page 815](#)

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### monitor traffic count

```
user@host> monitor traffic count 2
listening on fxp0
04:35:49.814125 In my-server.home.net.1295 > my-server.work.net.telnet: . ack
4122529478 win 16798 (DF)
04:35:49.814185
Out my-server.work.net.telnet > my-server.home.net.1295: P
1:38(37) ack 0 win 17680 (DF) [tos 0x10]
```

### monitor traffic detail count

```
user@host> monitor traffic detail count 2
listening on fxp0
04:38:16.265864 In my-server.home.net.1295 > my-server.work.net.telnet: . ack
4122529971 win 17678 (DF) (ttl 121, id 6812)
04:38:16.265926
Out my-server.work.net.telnet.telnet > my-server.home.net.1295: P 1:38(37) ack 0
win 17680 (DF) [tos 0x10] (ttl 6)
```

### monitor traffic extensive (Absolute Sequence)

```
user@host> monitor traffic extensive no-domain-names no-resolve no-timestamp count 20
matching "tcp" absolute-sequence
listening on fxp0
In 207.17.136.193.179 > 192.168.4.227.1024: . 4042780859:4042780859(0)
ack 1845421797 win 16384 <nop,nop,timestamp 4935628 965951> [tos 0xc0] (ttl )
In 207.17.136.193.179 > 192.168.4.227.1024: P 4042780859:4042780912(53)
ack 1845421797 win 16384
<nop,nop,timestamp 4935628 965951>:
BGP [|BGP UPDAT)
In 192.168.4.227.1024 > 207.17.136.193.179:
P 1845421797:1845421852(55) ack 4042780912 win 16384 <nop,nop,timestamp 965951
4935628>: BGP [|BGP UPDAT)
...
```

### monitor traffic extensive (Relative Sequence)

```
user@host> monitor traffic extensive no-domain-names no-resolve no-timestamp count 20
matching "tcp"
listening on fxp0
In 172.24.248.221.1680 > 192.168.4.210.23: . 396159737:396159737(0)
ack 1664980689 win 17574 (DF) (ttl 121, id 50003)
Out 192.168.4.210.23 > 172.24.248.221.1680: P 1:40(39)
ack 0 win 17680 (DF) [tos 0x10] (ttl 64, id 5394)
In 207.17.136.193.179 > 192.168.4.227.1024: P 4042775817:4042775874(57)
ack 1845416593 win 16384 <nop,nop,timestamp 4935379 965690>: BGP [|BGP UPDAT)
...
```

### monitor traffic extensive count

```
user@host> monitor traffic extensive count 5 no-domain-names no-resolve
listening on fxp013:18:17.406933
In 192.168.4.206.2723610880 > 172.17.28.8.2049:
40 null (ttl 64, id 38367)13:18:17.407577
In 172.17.28.8.2049 > 192.168.4.206.2723610880:
```

```

reply ok 28 null (ttl 61, id 35495)13:18:17.541140
In 0:e0:1e:42:9c:e0 0:e0:1e:42:9c:e0 9000 60:
0000 0100 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 0000 0000
0000 0000 000013:18:17.591513
In 172.24.248.156.4139 > 192.168.4.210.23:
3556964918:3556964918(0)
ack 295526518 win 17601 (DF)
(ttl 121, id 14)13:18:17.591568
Out 192.168.4.210.23 >
172.24.248.156.4139: P 1:40(39)
ack 0 win 17680 (DF) [tos 0x10]
(ttl 64, id 52376)

```

### monitor traffic interface

```

user@host> monitor traffic interface fxp0
listening on fxp0.0
18:17:28.800650 In server.home.net.723 > host1-0.lab.home.net.log
18:17:28.800733 Out host2-0.lab.home.net.login > server.home.net.7
18:17:28.817813 In host30.lab.home.net.syslog > host40.home0
18:17:28.817846 In host30.lab.home.net.syslog > host40.home0
...

```

### monitor traffic matching

```

user@host> monitor traffic matching "net 192.168.1.0/24"
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on fxp0, capture size 96 bytes

Reverse lookup for 192.168.1.255 failed (check DNS reachability).
Other reverse lookup failures will not be reported.
Use no-resolve to avoid reverse lookups on IP addresses.

21:55:54.003511 In IP truncated-ip - 18 bytes missing!
192.168.1.17.netbios-ns > 192.168.1.255.netbios-ns: UDP, length 50
21:55:54.003585 Out IP truncated-ip - 18 bytes missing!
192.168.1.17.netbios-ns > 192.168.1.255.netbios-ns: UDP, length 50
21:55:54.003864 In arp who-has 192.168.1.17 tell 192.168.1.9
...

```

### monitor traffic (TX Matrix Plus Router)

```

user@host> monitor traffic
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on em0, capture size 96 bytes
04:11:59.862121 Out IP truncated-ip - 25 bytes missing!
summit-em0.englab.juniper.net.syslog > sv-log-01.englab.juniper.net.syslog:
SYSLOG kernel.info, length: 57
04:11:59.862303
Out IP truncated-ip - 25 bytes missing!
summit-em0.englab.juniper.net.syslog >
sv-log-02.englab.juniper.net.syslog: SYSLOG kernel.info, length: 57
04:11:59.923948
In IP aj-em0.englab.juniper.net.65235 >

```

```

summit-em0.englab.juniper.net.telnet: .
ack 1087492766 win 33304 <nop,nop,timestamp 42366734 993490>
04:11:59.923983 Out IP truncated-ip - 232 bytes missing!
summit-em0.englab.juniper.net.telnet > aj-em0.englab.juniper.net.65235: P
1:241(240) ack 0 win 33304
<nop,nop,timestamp 993590 42366734>
04:12:00.022900
In IP aj-em0.englab.juniper.net.65235 >
summit-em0.englab.juniper.net.telnet: . ack 241 win 33304 <nop,nop,timestamp
42366834 993590>
04:12:00.141204
In IP truncated-ip - 40 bytes missing!
ipg-lnx-shell1.juniper.net.46182 > summit-em0.englab.juniper.net.telnet: P
2950530356:2950530404(48) ack 485494987 win 63712
<nop,nop,timestamp 1308555294 987086>
04:12:00.141345
Out IP summit-em0.englab.juniper.net.telnet >
ipg-lnx-shell1.juniper.net.46182: P 1:6(5)
ack 48 win 33304
<nop,nop,timestamp 993809 1308555294>
04:12:00.141572
In IP ipg-lnx-shell1.juniper.net.46182 >
summit-em0.englab.juniper.net.telnet: .
ack 6 win 63712
<nop,nop,timestamp 1308555294 993809>
04:12:00.141597
Out IP summit-em0.englab.juniper.net.telnet >
ipg-lnx-shell1.juniper.net.46182: P 6:10(4) ack 48 win 33304
<nop,nop,timestamp 993810 1308555294>
04:12:00.141821
In IP ipg-lnx-shell1.juniper.net.46182 >
summit-em0.englab.juniper.net.telnet: .
ack 10 win 63712 <nop,nop,timestamp 1308555294 993810>
04:12:00.141837 Out IP truncated-ip - 2 bytes missing!
summit-em0.englab.juniper.net.telnet >
ipg-lnx-shell1.juniper.net.46182: P 10:20(10) ack 48 win 33304
<nop,nop,timestamp 993810 1308555294>
04:12:00.142072
In IP ipg-lnx-shell1.juniper.net.46182 >
summit-em0.englab.juniper.net.telnet: . ack 20 win 63712
<nop,nop,timestamp 1308555294 993810>
04:12:00.142089 Out IP summit-em0.englab.juniper.net.telnet >
ipg-lnx-shell1.juniper.net.46182: P 20:28(8) ack 48 win 33304 <nop,nop,timestamp
993810 1308555294>
04:12:00.142321
In IP ipg-lnx-shell1.juniper.net.46182 >
summit-em0.englab.juniper.net.telnet: .
ack 28 win 63712 <nop,nop,timestamp 1308555294 993810>
04:12:00.142337
Out IP truncated-ip - 1 bytes missing!
summit-em0.englab.juniper.net.telnet >
ipg-lnx-shell1.juniper.net.46182: P 28:37(9) ack 48 win 33304 <nop,nop,timestamp
993810 1308555294>
...

```

### monitor traffic (QFX3500 Switch)

```

user@switch> monitor traffic
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.

```

```
Listening on me4, capture size 96 bytes
Reverse lookup for 172.22.16.246 failed (check DNS reachability).
Other reverse lookup failures will not be reported.
Use <no-resolve> to avoid reverse lookups on IP addresses.
16:35:32.240873 Out IP truncated-ip - 112 bytes missing!
labqfx-me0.lab4.juniper.net.ssh >
172.22.16.246.telefinder: P 4200727624:4200727756(132) ack 2889954831 win 65535
16:35:32.240900 Out IP truncated-ip - 176 bytes missing!
labqfx-me0.lab4.juniper.net.ssh >
172.22.16.246.telefinder: P 132:328(196) ack 1 win 65535
...
```

## request validation policy

<b>Syntax</b>	request validation policy <instance <i>instance-name</i> > <logical-system <i>logical-system-name</i> > <record <i>ip-prefix</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	When BGP origin validation is configured, manually request a route validation record policy to be reevaluated. This command causes dependent route validation records to be reevaluated. Dependent route validation records are exactly matching and more specific records.
<b>Options</b>	<p><b>none</b>—Request a policy reevaluation of all dependent route validation records.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Request a policy reevaluation of all dependent route validation records for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p> <p><b>record <i>ip-prefix</i></b>—(Optional) Request a policy reevaluation of all route validation records that match a given prefix.</p>
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">request validation policy on page 817</a>
<b>Output Fields</b>	When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### request validation policy

```
user@host> request validation policy
  Enqueued 1 IPv4 records
  Enqueued 0 IPv6 records
```

## restart

### List of Syntax [Syntax on page 818](#)

[Syntax \(ACX Series Routers\) on page 818](#)  
[Syntax \(EX Series Switches\) on page 818](#)  
[Syntax \(Routing Matrix\) on page 819](#)  
[Syntax \(J Series Routing Platform\) on page 819](#)  
[Syntax \(TX Matrix Routers\) on page 819](#)  
[Syntax \(TX Matrix Plus Routers\) on page 819](#)  
[Syntax \(MX Series Routers\) on page 819](#)  
[Syntax \(J Series Routers\) on page 820](#)  
[Syntax \(QFX Series\) on page 820](#)

### Syntax 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 logical-system-name> | sampling
| sbc-configuration-process | sdk-service | service-deployment | services | services pgcp
gateway gateway-name | snmp | soft | static-subscribers | statistics-service |
subscriber-management | subscriber-management-helper | tunnel-oamd | usb-control |
vrrp | web-management>
<gracefully | immediately | soft>
```

### Syntax (ACX Series Routers)

```
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>
```

### Syntax (EX Series Switches)

```
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   mounstd-service   multicast-snooping   pgm   redundancy-interface-process   remote-operations   routing   secure-neighbor-discovery   service-deployment   sflow-service   snmp   vrrp   web-management>
<b>Syntax (Routing Matrix)</b>	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> > <gracefully   immediately   soft>
<b>Syntax (J Series Routing Platform)</b>	restart <adaptive-services   audit-process   chassis-control   class-of-service   dhcp   dialer-services   dls   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> <gracefully   immediately   soft>
<b>Syntax (TX Matrix Routers)</b>	restart <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> <all-chassis   all-lcc   lcc <i>number</i>   scc> <gracefully   immediately   soft>
<b>Syntax (TX Matrix Plus Routers)</b>	restart <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> <all-chassis   all-lcc   all-sfc   lcc <i>number</i>   sfc <i>number</i> > <gracefully   immediately   soft>
<b>Syntax (MX Series Routers)</b>	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   mounstd-service   mpls-traceroute   msp   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 > <all-members> <gracefully   immediately   soft> <local> <member <i>member-id</i> >
Syntax (J Series Routers)	restart <adaptive-services   audit-process   chassis-control   class-of-service   dhcp   dhcp-service   dialer-services   diameter-service   dlsr   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 > <gracefully   immediately   soft>
Syntax (QFX Series)	restart <adaptive-services   audit-process   chassis-control   class-of-service   dialer-services   diameter-service   dlsr   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   <i>logical-system-name</i> >   routing   sampling   secure-neighbor-discovery   service-deployment   snmp   usb-control   web-management > <gracefully   immediately   soft>
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 11.1 for the QFX Series. Command introduced in Junos OS Release 12.2 for ACX Series routers. Options added: <ul style="list-style-type: none"> <li>• <b>dynamic-flow-capture</b> in Junos OS Release 7.4.</li> <li>• <b>dlsr</b> in Junos OS Release 7.5.</li> <li>• <b>event-processing</b> in Junos OS Release 7.5.</li> <li>• <b>ppp</b> in Junos OS Release 7.5.</li> <li>• <b>l2ald</b> in Junos OS Release 8.0.</li> <li>• <b>link-management</b> in Release 8.0.</li> <li>• <b>pgcp-service</b> in Junos OS Release 8.4.</li> <li>• <b>sbc-configuration-process</b> in Junos OS Release 9.5.</li> <li>• <b>services pgcp gateway</b> in Junos OS Release 9.6.</li> <li>• <b>sfc</b> and <b>all-sfc</b> for the TX Matrix Router in Junos OS Release 9.6.</li> </ul>

**Description** Restart a Junos OS process.



**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.

**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.

**dapath-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.

**dlsw**—(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 router that is connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

**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.

**sbc-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 827](#)

**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 bfd session

<b>List of Syntax</b>	<a href="#">Syntax on page 829</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 829</a>
<b>Syntax</b>	<pre>show bfd session &lt;brief   detail   extensive   summary&gt; &lt;address address&gt; &lt;client rsvp-oam (brief   detail   extensive   summary)   vpls-oam (brief   detail   extensive   instance instance-name   summary)&gt; &lt;discriminator discriminator&gt; &lt;logical-system (all   logical-system-name)&gt; &lt;prefix address&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>show bfd session &lt;brief   detail   extensive   summary&gt; &lt;address address&gt; &lt;client rsvp-oam (brief   detail   extensive   summary)   vpls-oam (brief   detail   extensive   instance instance-name   summary)&gt; &lt;discriminator discriminator&gt; &lt;prefix address&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Options <b>discriminator</b> and <b>address</b> introduced in Junos OS Release 8.2.</p> <p>Option <b>prefix</b> introduced in Junos OS Release 9.0.</p> <p>Command introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Option <b>client</b> introduced in Junos OS Release 12.3R3.</p>
<b>Description</b>	Display information about active Bidirectional Forwarding Detection (BFD) sessions.
<b>Options</b>	<p><b>none</b>—(Same as <b>brief</b>) Display information about active BFD sessions.</p> <p><b>brief   detail   extensive   summary</b>—(Optional) Display the specified level of output.</p> <p><b>address address</b>—(Optional) Display information about the BFD session for the specified neighbor address.</p> <p><b>client rsvp-oam</b>  <b>(brief   detail   extensive   summary)</b>  <b>  vpls-oam</b>  <b>(brief   detail   extensive   instance instance-name   summary)</b>—(Optional) Display information about RSVP-OAM or VPLS-OAM BFD sessions in the specified level of output. For VPLS-OAM, display the specified level of output or display information about all of the BFD sessions for the specified VPLS routing instance.</p> <p><b>discriminator discriminator</b>—(Optional) Display information about the BFD session using the specified local discriminator.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>

**prefix address**—(Optional) Display information about all of the BFD sessions for the specified LDP forwarding equivalence class (FEC).

**Required Privilege Level** view

**Related Documentation**

- [clear bfd session on page 798](#)
- *Examples: Configuring BFD for Static Routes*
- *Example: Configuring BFD for OSPF*
- [Example: Configuring BFD for BGP on page 349](#)
- *Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol*
- *Example: Configuring BFD for IS-IS*

**List of Sample Output**

- [show bfd session on page 833](#)
- [show bfd session brief on page 834](#)
- [show bfd session detail on page 834](#)
- [show bfd session detail \(with Authentication\) on page 834](#)
- [show bfd session address extensive on page 834](#)
- [show bfd session client rsvp-oam on page 835](#)
- [show bfd session client vpls-oam summary on page 835](#)
- [show bfd session client vpls-oam instance instance-name on page 835](#)
- [show bfd session extensive on page 835](#)
- [show bfd session extensive \(with Authentication\) on page 836](#)
- [show bfd session summary on page 836](#)

**Output Fields** Table 12 on page 830 describes the output fields for the **show bfd session** command. Output fields are listed in the approximate order in which they appear.

**Table 12: show bfd session Output Fields**

Field Name	Field Description	Level of Output
Address	Address on which the BFD session is active.	brief detail extensive none
State	State of the BFD session: <b>Up</b> , <b>Down</b> , <b>Init</b> (initializing), or <b>Failing</b> .	brief detail extensive none
Interface	Interface on which the BFD session is active.	brief detail extensive none
Detect Time	Negotiated time interval, in seconds, used to detect BFD control packets.	brief detail extensive none
Transmit Interval	Time interval, in seconds, used by the transmitting system to send BFD control packets.	brief detail extensive none
Multiplier	Negotiated multiplier by which the time interval is multiplied to determine the detection time for the transmitting system.	detail extensive

Table 12: show bfd session Output Fields (*continued*)

Field Name	Field Description	Level of Output
Session up time	How long a BFD session has been established.	detail extensive
Client	Protocol for which the BFD session is active: <b>ISIS</b> , <b>OSPF</b> , or <b>Static</b> .	detail extensive
TX interval	Time interval, in seconds, used by the host system to transmit BFD control packets.	brief detail extensive none
RX interval	Time interval, in seconds, used by the host system to receive BFD control packets.	brief detail extensive none
Authenticate	Indicates that BFD authentication is configured.	detail extensive
keychain	Name of the security authentication keychain being used by a specific client.  BFD authentication information for a client is provided in a single line and includes the <b>keychain</b> , <b>algo</b> , and <b>mode</b> parameters. Multiple clients can be configured on a BFD session.	extensive
algo	BFD authentication algorithm being used for a specific client: <b>keyed-md5</b> , <b>keyed-sha-1</b> , <b>meticulous-keyed-md5</b> , <b>meticulous-keyed-sha-1</b> , or <b>simple-password</b> .  BFD authentication information for a client is provided in a single line and includes the <b>keychain</b> , <b>algo</b> , and <b>mode</b> parameters. Multiple clients can be configured on a BFD session.	extensive
mode	Level of BFD authentication enforcement being used by a specific client: <b>strict</b> or <b>loose</b> . Strict enforcement indicates that authentication is configured at both ends of the session (the default). Loose enforcement indicates that one end of the session might not be authenticated.  BFD authentication information for a client is provided in a single line and includes the <b>keychain</b> , <b>algo</b> , and <b>mode</b> parameters. Multiple clients can be configured on a BFD session.	extensive
Local diagnostic	Local diagnostic information about failing BFD sessions.	detail extensive
Remote diagnostic	Remote diagnostic information about failing BFD sessions.	detail extensive
Remote state	Reports whether the remote system's BFD packets have been received and whether the remote system is receiving transmitted control packets.	detail extensive
Version	BFD version: <b>0</b> or <b>1</b> .	extensive
Replicated	The <b>replicated</b> flag appears when nonstop routing or graceful Routing Engine switchover is configured and the BFD session has been replicated to the backup Routing Engine.	detail extensive
Min async interval	Minimum amount of time, in seconds, between asynchronous control packet transmissions across the BFD session.	extensive

Table 12: show bfd session Output Fields (*continued*)

Field Name	Field Description	Level of Output
Min slow interval	Minimum amount of time, in seconds, between synchronous control packet transmissions across the BFD session.	extensive
Adaptive async TX interval	Transmission interval being used because of adaptation.	extensive
RX interval	Minimum required receive interval.	extensive
Local min TX interval	Minimum amount of time, in seconds, between control packet transmissions on the local system.	extensive
Local min RX interval	Minimum amount of time, in seconds, between control packet detections on the local system.	extensive
Remote min TX interval	Minimum amount of time, in seconds, between control packet transmissions on the remote system.	extensive
Remote min RX interval	Minimum amount of time, in seconds, between control packet detections on the remote system.	extensive
Threshold transmission interval	Threshold for notification if the transmission interval increases.	extensive
Threshold for detection time	Threshold for notification if the detection time increases.	extensive
Local discriminator	Authentication code used by the local system to identify that BFD session.	extensive
Remote discriminator	Authentication code used by the remote system to identify that BFD session.	extensive
Echo mode	Information about the state of echo transmissions on the BFD session.	extensive
Prefix	LDP FEC address associated with the BFD session.	All levels
Egress, Destination	Displays the LDP FEC destination address. This field is displayed only on a router at the egress of an LDP FEC, where the BFD session has an LDP Operation, Administration, and Maintenance (OAM) client.	All levels
Remote is control-plane independent	<p>The BFD session on the remote peer is running on its Packet Forwarding Engine. In this case, when the remote node undergoes a graceful restart, the local peer can help the remote peer with the graceful restart.</p> <p>The following BFD sessions are not distributed to the Packet Forwarding Engine: tunnel-encapsulated sessions, and sessions over integrated routing and bridging (IRB) interfaces.</p>	extensive

Table 12: show bfd session Output Fields (*continued*)

Field Name	Field Description	Level of Output
Authentication	<p>Summary status of BFD authentication:</p> <ul style="list-style-type: none"> <li><b>status</b>—<b>enabled/active</b> indicates authentication is configured and active. <b>enabled/inactive</b> indicates authentication is configured but not active. This only occurs when the remote end of the session does not support authentication and loose checking is configured.</li> <li><b>keychain</b>—Name of the security authentication keychain associated with the specified BFD session.</li> <li><b>algo</b>—BFD authentication algorithm being used: <b>keyed-md5</b>, <b>keyed-sha-1</b>, <b>meticulous-keyed-md5</b>, <b>meticulous-keyed-sha-1</b>, or <b>simple-password</b>.</li> <li><b>mode</b>—Level of BFD authentication enforcement: <b>strict</b> or <b>loose</b>. Strict enforcement indicates authentication is configured at both ends of the session (the default). Loose enforcement indicates that one end of the session might not be authenticated.</li> </ul> <p>This information is only shown if BFD authentication is configured.</p>	<b>extensive</b>
Session ID	The BFD session ID number that represents the protection using MPLS fast reroute (FRR) and loop-free alternate (LFA).	<b>detail extensive</b>
sessions	Total number of active BFD sessions.	All levels
clients	Total number of clients that are hosting active BFD sessions.	All levels
Cumulative transmit rate	Total number of BFD control packets transmitted per second on all active sessions.	All levels
Cumulative receive rate	Total number of BFD control packets received per second on all active sessions.	All levels
Multi-hop, min-recv-TTL	Minimum time to live (TTL) accepted if the session is configured for multihop.	<b>extensive</b>
route table	Route table used if the session is configured for multihop.	<b>extensive</b>
local address	<p>Local address of the source used if the session is configured for multihop.</p> <p>The source IP address for outgoing BFD packets from the egress side of an MPLS BFD session is based on the outgoing interface IP address.</p>	<b>extensive</b>

## Sample Output

### show bfd session

```
user@host> show bfd session
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.9.1.33	Up	so-7/1/0.0	0.600	0.200	3
10.9.1.29	Up	ge-4/0/0.0	0.600	0.200	3

```
2 sessions, 2 clients
```

```
Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps
```

### show bfd session brief

The output for the **show bfd session brief** command is identical to that for the **show bfd session** command. For sample output, see [show bfd session on page 833](#).

### show bfd session detail

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

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.9.1.33	Up	so-7/1/0.0	0.600	0.200	3
Client OSPF, TX interval 0.200, RX interval 0.200, multiplier 3					
Session up time 3d 00:34					
Local diagnostic None, remote diagnostic None					
Remote state Up, version 1					
Replicated					
10.9.1.29	Up	ge-4/0/0.0	0.600	0.200	3
Client ISIS L2, TX interval 0.200, RX interval 0.200, multiplier 3					
Session up time 3d 00:29, previous down time 00:00:01					
Local diagnostic NbrSignal, remote diagnostic AdminDown					
Remote state Up, version 1					

2 sessions, 2 clients

Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps

### show bfd session detail (with Authentication)

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

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.9.1.33	Up	so-7/1/0.0	0.600	0.200	3
Client OSPF, TX interval 0.200, RX interval 0.200, multiplier 3, <b>Authenticate</b>					
Session up time 3d 00:34					
Local diagnostic None, remote diagnostic None					
Remote state Up, version 1					
Replicated					
10.9.1.29	Up	ge-4/0/0.0	0.600	0.200	3
Client ISIS L2, TX interval 0.200, RX interval 0.200, multiplier 3					
Session up time 3d 00:29, previous down time 00:00:01					
Local diagnostic NbrSignal, remote diagnostic AdminDown					
Remote state Up, version 1					

2 sessions, 2 clients

Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps

### show bfd session address extensive

```
user@host> show bfd session 10.255.245.212 extensive
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.255.245.212	Up		1.200	0.400	3
Client Static, TX interval 0.400, RX interval 0.400, multiplier 3					
Session up time 00:17:03, previous down time 00:00:14					
Local diagnostic CtlExpire, remote diagnostic NbrSignal					
Remote state Up, version 1					
Replicated					
Min async interval 0.400, min slow interval 1.000					
Adaptive async tx interval 0.400, rx interval 0.400					
Local min tx interval 0.400, min rx interval 0.400, multiplier 3					
Remote min tx interval 0.400, min rx interval 0.400, multiplier 3					

```

Threshold transmission interval 0.000, Threshold for detection time 0.000
Local discriminator 6, remote discriminator 16
Echo mode disabled/inactive
Multi-hop, min-recv-TTL 255, route-table 0, local-address 10.255.245.205

```

```

1 sessions, 1 clients
Cumulative transmit rate 2.5 pps, cumulative receive rate 2.5 pps

```

#### show bfd session client rsvp-oam

```

user@host> show bfd session client rsvp-oam

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
192.168.0.223	Up		540.000	180.000	3

```

1 Up sessions, 0 Down sessions
1 sessions, 1 clients
Cumulative transmit rate 0.0 pps, cumulative receive rate 0.0 pps

```

#### show bfd session client vpls-oam summary

```

user@host> show bfd session client vpls-oam summary
1 Up sessions, 1 Down sessions
2 sessions, 2 clients
Cumulative transmit rate 2.0 pps, cumulative receive rate 1.0 pps

```

#### show bfd session client vpls-oam instance instance-name

```

user@host> show bfd session client vpls-oam instance vpls

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
127.0.0.1	Up	ae9.0	3.000	1.000	3

```

1 Up Sessions, 0 Down Sessions
1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps

```

#### show bfd session extensive

```

user@host> show bfd session extensive
10.31.1.2 Up ge-2/1/8.0 0.030 0.010 3
Client OSPF realm ospf-v2 Area 0.0.0.0, TX interval 0.010, RX interval 0.010
Session up time 00:10:13
Local diagnostic None, remote diagnostic None
Remote state Up, version 1
Replicated
Min async interval 0.010, min slow interval 1.000
Adaptive async TX interval 0.010, RX interval 0.010
Local min TX interval 0.010, minimum RX interval 0.010, multiplier 3
Remote min TX interval 0.010, min RX interval 0.010, multiplier 3
Local discriminator 12, remote discriminator 4
Echo mode disabled/inactive
Remote is control-plane independent
Session ID: 0x201
Micro-BFD Session

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.31.2.2	Up	ge-2/1/4.0	0.030	0.010	3

```

Client OSPF realm ospf-v2 Area 0.0.0.0, TX interval 0.010, RX interval 0.010

```

```

Session up time 00:10:14
Local diagnostic None, remote diagnostic NbrSignal
Remote state Up, version 1
Replicated
Min async interval 0.010, min slow interval 1.000
Adaptive async TX interval 0.010, RX interval 0.010
Local min TX interval 0.010, minimum RX interval 0.010, multiplier 3
Remote min TX interval 0.010, min RX interval 0.010, multiplier 3
Local discriminator 13, remote discriminator 5
Echo mode disabled/inactive
Remote is control-plane independent
Session ID: 0x202

```

```

2 sessions, 2 clients
Cumulative transmit rate 200.0 pps, cumulative receive rate 200.0 pps

```

#### show bfd session extensive (with Authentication)

```

user@host> show bfd session extensive

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
192.168.208.26	Up	so-1/0/0.0	2.400	0.800	10

```

Client Static, TX interval 0.600, RX interval 0.600, Authenticate
    keychain bfd, algo keyed-md5, mode loose
Session up time 00:18:07
Local diagnostic None, remote diagnostic NbrSignal
Remote state Up, version 1
Replicated
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 10
Remote min TX interval 0.800, min RX interval 0.800, multiplier 3
Local discriminator 2, remote discriminator 3
Echo mode disabled/inactive
Authentication enabled/active, keychain bfd, algo keyed-md5, mode loose

1 sessions, 1 clients
Cumulative transmit rate 1.2 pps, cumulative receive rate 1.2 pps

```

#### show bfd session summary

```

user@host> show bfd session summary
2 sessions, 2 clients
Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps

```

## show bgp bmp

<b>Syntax</b>	<b>show bgp bmp</b>
<b>Release Information</b>	Command introduced in Junos OS Release 9.5. Command introduced in Junos OS Release 9.5 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series.
<b>Description</b>	Display information about the BGP Monitoring Protocol (BMP).
<b>Options</b>	This command has no options.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show bgp bmp on page 837</a>
<b>Output Fields</b>	<a href="#">Table 13 on page 837</a> lists the output fields for the <b>show bgp bmp</b> command. Output fields are listed in the approximate order in which they appear.

**Table 13: show bgp bmp Output Fields**

Field Name	Field Description
<b>BMP station address/port</b>	IP address and port number of the monitoring station to which BGP Monitoring Protocol (BMP) statistics are sent.
<b>BMP session state</b>	Status of the BMP session: <b>UP</b> or <b>DOWN</b> .
<b>Memory consumed by BMP</b>	Memory used by the active BMP session.
<b>Statistics timeout</b>	Amount of time, in seconds, between transmissions of BMP data to the monitoring station.
<b>Memory limit</b>	Threshold, in bytes, at which the routing device stops collecting BMP data.
<b>Memory-connect retry timeout</b>	Amount of time, in seconds, after which the routing device attempts to resume a BMP session that was ended after the configured memory threshold was exceeded.

## Sample Output

### show bgp bmp

```

user@host> show bgp bmp
  BMP station address/port: 172.24.24.157+5454
  BMP session state: DOWN
  Memory consumed by BMP: 0
  Statistics timeout: 15
  Memory limit: 10485760
  Memory connect retry timeout: 600

```



## show bgp group

<b>List of Syntax</b>	<a href="#">Syntax on page 839</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 839</a>
<b>Syntax</b>	<pre>show bgp group &lt;brief   detail   summary&gt; &lt;group-name&gt; &lt;exact-instance instance-name&gt; &lt;instance instance-name&gt; &lt;logical-system (all   logical-system-name)&gt; &lt;rtf&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>show bgp group &lt;brief   detail   summary&gt; &lt;group-name&gt; &lt;exact-instance instance-name&gt; &lt;instance instance-name&gt;</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>exact-instance</b> option introduced in Junos OS Release 11.4.</p>
<b>Description</b>	Display information about the configured BGP groups.
<b>Options</b>	<p><b>none</b>—Display group information about all BGP groups.</p> <p><b>brief   detail   summary</b>—(Optional) Display the specified level of output.</p> <p><b>group-name</b>—(Optional) Display group information for the specified group.</p> <p><b>exact-instance instance-name</b>—(Optional) Display information for the specified instance only.</p> <p><b>instance instance-name</b>—(Optional) Display information about BGP groups for all routing instances whose name begins with this string (for example, <b>cust1</b>, <b>cust11</b>, and <b>cust111</b> are all displayed when you run the <b>show bgp group instance cust1</b> command). The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>rtf</b>—(Optional) Display BGP group route targeting information.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show bgp group on page 843</a> <a href="#">show bgp group brief on page 843</a> <a href="#">show bgp group detail on page 844</a>

[show bgp group rtf detail on page 845](#)

[show bgp group summary on page 845](#)

**Output Fields** Table 14 on page 840 describes the output fields for the **show bgp group** command. Output fields are listed in the approximate order in which they appear.

**Table 14: show bgp group Output Fields**

Field Name	Field Description	Level of Output
<b>Group Type or Group</b>	Type of BGP group: <b>Internal</b> or <b>External</b> .	All levels
<b>group-index</b>	Index number for the BGP peer group. The index number differentiates between groups when a single BGP group is split because of different configuration options at the group and peer levels.	<b>rtf detail</b>
<b>AS</b>	AS number of the peer. For internal BGP (IBGP), this number is the same as <b>Local AS</b> .	<b>brief detail</b> none
<b>Local AS</b>	AS number of the local routing device.	<b>brief detail</b> none
<b>Name</b>	Name of a specific BGP group.	<b>brief detail</b> none
<b>Index</b>	Unique index number of a BGP group.	<b>brief detail</b> none
<b>Flags</b>	Flags associated with the BGP group. This field is used by Juniper Networks customer support.	<b>brief detail</b> none
<b>Remove-private options</b>	Options associated with the <a href="#">remove-private</a> statement.	<b>brief detail</b> none
<b>Holdtime</b>	Maximum number of seconds allowed to elapse between successive keepalive or update messages that BGP receives from a peer in the BGP group, after which the connection to the peer is closed and routing devices through that peer become unavailable.	<b>brief detail</b> none
<b>Export</b>	Export policies configured for the BGP group with the <b>export</b> statement.	<b>brief detail</b> none
<b>MED tracks IGP metric update delay</b>	Time, in seconds, that updates to multiple exit discriminator (MED) are delayed. Also displays the time remaining before the interval is set to expire	All levels
<b>Traffic Statistics Interval</b>	Time between sample periods for labeled-unicast traffic statistics, in seconds.	<b>brief detail</b> none
<b>Total peers</b>	Total number of peers in the group.	<b>brief detail</b> none

Table 14: show bgp group Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Established</b>	Number of peers in the group that are in the established state.	All levels
<b>Active/Received/Accepted/Damped</b>	<p>Multipurpose field that displays information about BGP peer sessions. The field's contents depend upon whether a session is established and whether it was established in the main routing device or in a routing instance.</p> <ul style="list-style-type: none"> <li>If a peer is not established, the field shows the state of the peer session: <b>Active</b>, <b>Connect</b>, or <b>Idle</b>.</li> <li>If a BGP session is established in the main routing device, the field shows the number of active, received, accepted, and damped routes that are received from a neighbor and appear in the <b>inet.0</b> (main) and <b>inet.2</b> (multicast) routing tables. For example, <b>8/10/10/2</b> and <b>2/4/4/0</b> indicate the following: <ul style="list-style-type: none"> <li>8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the <b>inet.0</b> routing table.</li> <li>2 active routes, 4 received routes, 4 accepted routes, and no damped routes from a BGP peer appear in the <b>inet.2</b> routing table.</li> </ul> </li> </ul>	<b>summary</b>
<b>ip-addresses</b>	List of peers who are members of the group. The address is followed by the peer's port number.	All levels
<b>Route Queue Timer</b>	Number of seconds until queued routes are sent. If this time has already elapsed, this field displays the number of seconds by which the updates are delayed.	<b>detail</b>
<b>Route Queue</b>	Number of prefixes that are queued up for sending to the peers in the group.	<b>detail</b>
<b>inet.number</b>	<p>Number of active, received, accepted, and damped routes in the routing table. For example, <b>inet.0: 7/10/9/0</b> indicates the following:</p> <ul style="list-style-type: none"> <li>7 active routes, 10 received routes, 9 accepted routes, and no damped routes from a BGP peer appear in the <b>inet.0</b> routing table.</li> </ul>	none

Table 14: show bgp group Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Table inet.number</b>	Information about the routing table. <ul style="list-style-type: none"> <li>• <b>Received prefixes</b>—Total number of prefixes from the peer, both active and inactive, that are in the routing table.</li> <li>• <b>Active prefixes</b>—Number of prefixes received from the peer that are active in the routing table.</li> <li>• <b>Suppressed due to damping</b>—Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.</li> <li>• <b>Advertised prefixes</b>—Number of prefixes advertised to a peer.</li> <li>• <b>Received external prefixes</b>—Total number of prefixes from the external BGP (EBGP) peers, both active and inactive, that are in the routing table.</li> <li>• <b>Active external prefixes</b>—Number of prefixes received from the EBGP peers that are active in the routing table.</li> <li>• <b>Externals suppressed</b>—Number of routes received from EBGP peers currently inactive because of damping or other reasons.</li> <li>• <b>Received internal prefixes</b>—Total number of prefixes from the IBGP peers, both active and inactive, that are in the routing table.</li> <li>• <b>Active internal prefixes</b>—Number of prefixes received from the IBGP peers that are active in the routing table.</li> <li>• <b>Internals suppressed</b>—Number of routes received from IBGP peers currently inactive because of damping or other reasons.</li> <li>• <b>RIB State</b>—Status of the graceful restart process for this routing table: <b>BGP restart is complete</b>, <b>BGP restart in progress</b>, <b>VPN restart in progress</b>, or <b>VPN restart is complete</b>.</li> </ul>	<b>detail</b>
<b>Groups</b>	Total number of groups.	All levels
<b>Peers</b>	Total number of peers.	All levels
<b>External</b>	Total number of external peers.	All levels
<b>Internal</b>	Total number of internal peers.	All levels
<b>Down peers</b>	Total number of unavailable peers.	All levels
<b>Flaps</b>	Total number of flaps that occurred.	All levels
<b>Table</b>	Name of a routing table.	<b>brief</b> , none
<b>Tot Paths</b>	Total number of routes.	<b>brief</b> , none
<b>Act Paths</b>	Number of active routes.	<b>brief</b> , none
<b>Suppressed</b>	Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.	<b>brief</b> , none

Table 14: show bgp group Output Fields (*continued*)

Field Name	Field Description	Level of Output
History	Number of withdrawn routes stored locally to keep track of damping history.	<b>brief, none</b>
Damp State	Number of active routes with a figure of merit greater than zero, but lower than the threshold at which suppression occurs.	<b>brief, none</b>
Pending	Routes being processed by the BGP import policy.	<b>brief, none</b>
Group	Group the peer belongs to in the BGP configuration.	<b>detail</b>
Receive mask	Mask of the received target included in the advertised route.	<b>detail</b>
Entries	Number of route entries received.	<b>detail</b>
Target	Route target that is to be passed by route-target filtering. If a route advertised from the provider edge (PE) routing device matches an entry in the route-target filter, the route is passed to the peer.	<b>detail</b>
Mask	Mask which specifies that the peer receive routes with the given route target.	<b>detail</b>

## Sample Output

### show bgp group

```

user@host> show bgp group
Groups: 2  Peers: 2   External: 0   Internal: 2   Down peers: 1   Flaps: 0
Table      Tot Paths  Act Paths  Suppressed   History Damp State   Pending

inet.0
          0         0         0           0         0         0

bgp.13vpn.0
          0         0         0           0         0         0

bgp.rtarget.0
          2         0         0           0         0         0

```

### show bgp group brief

```

user@host> show bgp group brief
Groups: 2  Peers: 2   External: 0   Internal: 2   Down peers: 1   Flaps: 0
Table      Tot Paths  Act Paths  Suppressed   History Damp State   Pending

inet.0
          0         0         0           0         0         0

bgp.13vpn.0
          0         0         0           0         0         0

bgp.rtarget.0
          2         0         0           0         0         0

```

## show bgp group detail

```

user@host> show bgp group detail
Group Type: Internal   AS: 1                      Local AS: 1
Name: ibgp             Index: 0                    Flags: <Export Eval>
Holdtime: 0
Total peers: 3         Established: 0
22.0.0.2
22.0.0.8
22.0.0.5

Groups: 1 Peers: 3   External: 0   Internal: 3   Down peers: 3   Flaps: 3
Table bgp.l3vpn.0
  Received prefixes:      0
  Accepted prefixes:      0
  Active prefixes:        0
  Suppressed due to damping: 0
  Received external prefixes: 0
  Active external prefixes: 0
  Externals suppressed:   0
  Received internal prefixes: 0
  Active internal prefixes: 0
  Internals suppressed:   0
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
Table bgp.mdt.0
  Received prefixes:      0
  Accepted prefixes:      0
  Active prefixes:        0
  Suppressed due to damping: 0
  Received external prefixes: 0
  Active external prefixes: 0
  Externals suppressed:   0
  Received internal prefixes: 0
  Active internal prefixes: 0
  Internals suppressed:   0
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
Table VPN-A.inet.0
  Received prefixes:      0
  Accepted prefixes:      0
  Active prefixes:        0
  Suppressed due to damping: 0
  Received external prefixes: 0
  Active external prefixes: 0
  Externals suppressed:   0
  Received internal prefixes: 0
  Active internal prefixes: 0
  Internals suppressed:   0
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
Table VPN-A.mdt.0
  Received prefixes:      0
  Accepted prefixes:      0
  Active prefixes:        0
  Suppressed due to damping: 0
  Received external prefixes: 0
  Active external prefixes: 0
  Externals suppressed:   0
  Received internal prefixes: 0
  Active internal prefixes: 0

```

```

Internals suppressed:      0
RIB State: BGP restart is complete
RIB State: VPN restart is complete

```

### show bgp group rtf detail

```

user@host> show bgp group rtf detail
Group: internal (group-index: 0)
  Receive mask: 00000002
  Table: bgp.rtarget.0
    Target
    100:100/64
    200:201/64
    Mask
    00000002
    (Group)
    Entries: 2
Group: internal (group-index: 1)
  Table: bgp.rtarget.0
    Target
    200:201/64
    Mask
    (Group)
    Entries: 1

```

### show bgp group summary

```

user@host> show bgp group summary
Group      Type      Peers      Established      Active/Received/Accepted/Damped
ibgp       Internal  3           0
Groups: 1  Peers: 3      External: 0      Internal: 3      Down peers: 3      Flaps: 3
  bgp.l3vpn.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  bgp.mdt.0        : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  VPN-A.inet.0     : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  VPN-A.mdt.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0

```

## show bgp group traffic-statistics

<b>Syntax</b>	show bgp group traffic-statistics <brief   detail> <group-name> <logical-system (all   <i>logical-system-name</i> )>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4.
<b>Description</b>	Display the traffic statistics for configured Border Gateway Protocol (BGP) groups.
<b>Options</b>	<p><b>none</b>—Display traffic statistics for all BGP groups.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output.</p> <p><b>group-name</b>—(Optional) Display BGP traffic statistics for only the specified group.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show bgp group traffic-statistics (Per-Group-Label Not Configured) on page 847</a> <a href="#">show bgp group traffic-statistics (Per-Group-Label Configured) on page 847</a>
<b>Output Fields</b>	Table 15 on page 846 describes the output fields for the <b>show bgp group traffic-statistics</b> command. Output fields are listed in the approximate order in which they appear.

**Table 15: show bgp group traffic-statistics Output Fields**

Field Name	Field Description
<b>Group name</b>	Name of a specific BGP group.
<b>Group Index</b>	Index number for the BGP group.
<b>NLRI</b>	Network layer reachability information (NLRI) indicating the source of the traffic statistics for the BGP group.
<b>FEC</b>	Forwarding equivalence classes (FECs) associated with the BGP group.
<b>Packets</b>	Number of packets sent through each FEC.
<b>Bytes</b>	Number of bytes transmitted through each FEC.
<b>EgressAS</b>	Autonomous system (AS) number of the egress router.
<b>AdvLabel</b>	Label associated with each FEC.

## Sample Output

### show bgp group traffic-statistics (Per-Group-Label Not Configured)

```

user@host> show bgp group traffic-statistics
Group Name: ext1      Group Index: 0      NLRI: inet-labeled-unicast
FEC                   Packets            Bytes            EgressAS        AdvLabel
10.255.245.55         0                  0                I                100224
10.255.245.57         0                  0                I                100240
100.101.0.0           550                48400            25               100256
100.102.0.0           550                48400            25               100256
100.103.0.0           550                48400            25               100272
100.104.0.0           550                48400            25               100272
192.168.25.0          0                  0                I                100288

Group Name: ext2      Group Index: 1      NLRI: inet-labeled-unicast
FEC                   Packets            Bytes            EgressAS        AdvLabel
10.255.245.55         0                  0                I                100224
10.255.245.57         0                  0                I                100240
100.101.0.0           550                48400            25               100256
100.102.0.0           550                48400            25               100256
100.103.0.0           550                48400            25               100272
100.104.0.0           550                48400            25               100272
192.168.25.0          0                  0                I                100288

```

### show bgp group traffic-statistics (Per-Group-Label Configured)

```

user@host> show bgp group traffic-statistics
Group Name: ext1      Group Index: 0      NLRI: inet-labeled-unicast
FEC                   Packets            Bytes            EgressAS        AdvLabel
10.255.245.55         0                  0                I                100384
10.255.245.57         0                  0                I                100400
100.101.0.0           101                8888             25               100416
100.102.0.0           101                8888             25               100416
100.103.0.0           0                  0                25               100432
100.104.0.0           0                  0                25               100432
192.168.25.0          0                  0                I                100448

Group Name: ext2      Group Index: 1      NLRI: inet-labeled-unicast
FEC                   Packets            Bytes            EgressAS        AdvLabel
10.255.245.55         0                  0                I                100304
10.255.245.57         0                  0                I                100320
100.101.0.0           0                  0                25               100336
100.102.0.0           0                  0                25               100336
100.103.0.0           101                8888             25               100352
100.104.0.0           101                8888             25               100352
192.168.25.0          0                  0                I                100368

```

## show bgp neighbor

---

<b>List of Syntax</b>	<a href="#">Syntax on page 848</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 848</a>
<b>Syntax</b>	<pre>show bgp neighbor &lt;exact-instance <i>instance-name</i>&gt; &lt;instance <i>instance-name</i>&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt; &lt;neighbor-address&gt; &lt;orf (detail   <i>neighbor-address</i>)&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>show bgp neighbor &lt;instance <i>instance-name</i>&gt; &lt;exact-instance <i>instance-name</i>&gt; &lt;neighbor-address&gt; &lt;orf (<i>neighbor-address</i>   detail)&gt;</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>orf</b> option introduced in Junos OS Release 9.2.</p> <p><b>exact-instance</b> option introduced in Junos OS Release 11.4.</p>
<b>Description</b>	Display information about BGP peers.
<b>Options</b>	<p><b>none</b>—Display information about all BGP peers.</p> <p><b>exact-instance <i>instance-name</i></b>—(Optional) Display information for the specified instance only.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information about BGP peers for all routing instances whose name begins with this string (for example, <b>cust1</b>, <b>cust11</b>, and <b>cust111</b> are all displayed when you run the <b>show bgp neighbor instance cust1</b> command).</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>neighbor-address</b>—(Optional) Display information for only the BGP peer at the specified IP address.</p> <p><b>orf (detail   <i>neighbor-address</i>)</b>—(Optional) Display outbound route-filtering information for all BGP peers or only for the BGP peer at the specified IP address. The default is to display brief output. Use the <b>detail</b> option to display detailed output.</p>
<b>Additional Information</b>	For information about the <b>local-address</b> , <b>nlri</b> , <b>hold-time</b> , and <b>preference</b> statements, see the <i>Junos OS Routing Protocols Library for Routing Devices</i> .
<b>Required Privilege Level</b>	view

**Related Documentation** • [clear bgp neighbor on page 800](#)

**List of Sample Output** [show bgp neighbor on page 855](#)  
[show bgp neighbor \(CLNS\) on page 856](#)  
[show bgp neighbor \(Layer 2 VPN\) on page 857](#)  
[show bgp neighbor \(Layer 3 VPN\) on page 859](#)  
[show bgp neighbor neighbor-address on page 859](#)  
[show bgp neighbor neighbor-address on page 860](#)  
[show bgp neighbor orf neighbor-address detail on page 861](#)

**Output Fields** [Table 16 on page 849](#) describes the output fields for the **show bgp neighbor** command. Output fields are listed in the approximate order in which they appear.

**Table 16: show bgp neighbor Output Fields**

Field Name	Field Description
<b>Peer</b>	Address of the BGP neighbor. The address is followed by the neighbor port number.
<b>AS</b>	AS number of the peer.
<b>Local</b>	Address of the local routing device. The address is followed by the peer port number.
<b>Type</b>	Type of peer: <b>Internal</b> or <b>External</b> .
<b>State</b>	<p>Current state of the BGP session:</p> <ul style="list-style-type: none"> <li>• <b>Active</b>—BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an Open message.</li> <li>• <b>Connect</b>—BGP is waiting for the transport protocol connection to be completed.</li> <li>• <b>Established</b>—The BGP session has been established, and the peers are exchanging update messages.</li> <li>• <b>Idle</b>—This is the first stage of a connection. BGP is waiting for a Start event.</li> <li>• <b>OpenConfirm</b>—BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message.</li> <li>• <b>OpenSent</b>—BGP has sent an open message and is waiting to receive an open message from the peer.</li> </ul>

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
<b>Flags</b>	<p>Internal BGP flags:</p> <ul style="list-style-type: none"> <li>• <b>Aggregate Label</b>—BGP has aggregated a set of incoming labels (labels received from the peer) into a single forwarding label.</li> <li>• <b>CleanUp</b>—The peer session is being shut down.</li> <li>• <b>Delete</b>—This peer has been deleted.</li> <li>• <b>Idled</b>—This peer has been permanently idled.</li> <li>• <b>ImportEval</b>—At the last commit operation, this peer was identified as needing to reevaluate all received routes.</li> <li>• <b>Initializing</b>—The peer session is initializing.</li> <li>• <b>SendRtn</b>—Messages are being sent to the peer.</li> <li>• <b>Sync</b>—This peer is synchronized with the rest of the peer group.</li> <li>• <b>TryConnect</b>—Another attempt is being made to connect to the peer.</li> <li>• <b>Unconfigured</b>—This peer is not configured.</li> <li>• <b>WriteFailed</b>—An attempt to write to this peer failed.</li> </ul>
<b>Last state</b>	<p>Previous state of the BGP session:</p> <ul style="list-style-type: none"> <li>• <b>Active</b>—BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an Open message.</li> <li>• <b>Connect</b>—BGP is waiting for the transport protocol connection to be completed.</li> <li>• <b>Established</b>—The BGP session has been established, and the peers are exchanging update messages.</li> <li>• <b>Idle</b>—This is the first stage of a connection. BGP is waiting for a Start event.</li> <li>• <b>OpenConfirm</b>—BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message.</li> <li>• <b>OpenSent</b>—BGP has sent an open message and is waiting to receive an open message from the peer.</li> </ul>
<b>Last event</b>	<p>Last activity that occurred in the BGP session:</p> <ul style="list-style-type: none"> <li>• <b>Closed</b>—The BGP session closed.</li> <li>• <b>ConnectRetry</b>—The transport protocol connection failed, and BGP is trying again to connect.</li> <li>• <b>HoldTime</b>—The session ended because the hold timer expired.</li> <li>• <b>KeepAlive</b>—The local routing device sent a BGP keepalive message to the peer.</li> <li>• <b>Open</b>—The local routing device sent a BGP open message to the peer.</li> <li>• <b>OpenFail</b>—The local routing device did not receive an acknowledgment of a BGP open message from the peer.</li> <li>• <b>RecvKeepAlive</b>—The local routing device received a BGP keepalive message from the peer.</li> <li>• <b>RecvNotify</b>—The local routing device received a BGP notification message from the peer.</li> <li>• <b>RecvOpen</b>—The local routing device received a BGP open message from the peer.</li> <li>• <b>RecvUpdate</b>—The local routing device received a BGP update message from the peer.</li> <li>• <b>Start</b>—The peering session started.</li> <li>• <b>Stop</b>—The peering session stopped.</li> <li>• <b>TransportError</b>—A TCP error occurred.</li> </ul>

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
Last error	<p>Last error that occurred in the BGP session:</p> <ul style="list-style-type: none"> <li>• <b>Cease</b>—An error occurred, such as a version mismatch, that caused the session to close.</li> <li>• <b>Finite State Machine Error</b>—In setting up the session, BGP received a message that it did not understand.</li> <li>• <b>Hold Time Expired</b>—The session's hold time expired.</li> <li>• <b>Message Header Error</b>—The header of a BGP message was malformed.</li> <li>• <b>Open Message Error</b>—A BGP open message contained an error.</li> <li>• <b>None</b>—No errors occurred in the BGP session.</li> <li>• <b>Update Message Error</b>—A BGP update message contained an error.</li> </ul>
Export	Name of the export policy that is configured on the peer.
Import	Name of the import policy that is configured on the peer.
Options	<p>Configured BGP options:</p> <ul style="list-style-type: none"> <li>• <b>AddressFamily</b>—Configured address family: <b>inet</b> or <b>inet-vpn</b>.</li> <li>• <b>AuthKeyChain</b>—Authentication key change is enabled.</li> <li>• <b>DropPathAttributes</b>—Certain path attributes are configured to be dropped from neighbor updates during inbound processing.</li> <li>• <b>GracefulRestart</b>—Graceful restart is configured.</li> <li>• <b>HoldTime</b>—Hold time configured with the <b>hold-time</b> statement. The hold time is three times the interval at which keepalive messages are sent.</li> <li>• <b>IgnorePathAttributes</b>—Certain path attributes are configured to be ignored in neighbor updates during inbound processing.</li> <li>• <b>Local Address</b>—Address configured with the <b>local-address</b> statement.</li> <li>• <b>Multihop</b>—Allow BGP connections to external peers that are not on a directly connected network.</li> <li>• <b>NLRI</b>—Configured MBGP state for the BGP group: <b>multicast</b>, <b>unicast</b>, or both if you have configured <b>nlri any</b>.</li> <li>• <b>Peer AS</b>—Configured peer autonomous system (AS).</li> <li>• <b>Preference</b>—Preference value configured with the <b>preference</b> statement.</li> <li>• <b>Refresh</b>—Configured to refresh automatically when the policy changes.</li> <li>• <b>Rib-group</b>—Configured routing table group.</li> </ul>
Path-attributes dropped	Path attribute codes that are dropped from neighbor updates.
Path-attributes ignored	Path attribute codes that are ignored during neighbor updates.
Authentication key change	(appears only if the <b>authentication-keychain</b> statement has been configured) Name of the authentication keychain enabled.
Authentication algorithm	(appears only if the <b>authentication-algorithm</b> statement has been configured) Type of authentication algorithm enabled: <b>hmac</b> or <b>md5</b> .
Address families configured	Names of configured address families for the VPN.

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
Local Address	Address of the local routing device.
Remove-private options	Options associated with the <code>remove-private</code> statement.
Holdtime	Hold time configured with the <code>hold-time</code> statement. The hold time is three times the interval at which keepalive messages are sent.
Flags for NLRI inet-label-unicast	Flags related to labeled-unicast: <ul style="list-style-type: none"> <li>• <b>TrafficStatistics</b>—Collection of statistics for labeled-unicast traffic is enabled.</li> </ul>
Traffic statistics	Information about labeled-unicast traffic statistics: <ul style="list-style-type: none"> <li>• <b>Options</b>—Options configured for collecting statistics about labeled-unicast traffic.</li> <li>• <b>File</b>—Name and location of statistics log files.</li> <li>• <b>size</b>—Size of all the log files, in bytes.</li> <li>• <b>files</b>—Number of log files.</li> </ul>
Traffic Statistics Interval	Time between sample periods for labeled-unicast traffic statistics, in seconds.
Preference	Preference value configured with the <code>preference</code> statement.
Outbound Timer	Time for which the route is available in Junos OS routing table before it is exported to BGP. This field is displayed in the output only if the <code>out-delay</code> parameter is configured to a non-zero value.
Number of flaps	Number of times the BGP session has gone down and then come back up.
Peer ID	Router identifier of the peer.
Group index	Index number for the BGP peer group. The index number differentiates between groups when a single BGP group is split because of different configuration options at the group and peer levels.
Peer index	Index that is unique within the BGP group to which the peer belongs.
Local ID	Router identifier of the local routing device.
Local Interface	Name of the interface on the local routing device.
Active holdtime	Hold time that the local routing device negotiated with the peer.
Keepalive Interval	Keepalive interval, in seconds.
BFD	Status of BFD failure detection.
Local Address	Name of directly connected interface over which direct EBGP peering is established.
NLRI for restart configured on peer	Names of address families configured for restart.

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
NLRI advertised by peer	Address families supported by the peer: <b>unicast</b> or <b>multicast</b> .
NLRI for this session	Address families being used for this session.
Peer supports Refresh capability	Remote peer's ability to send and request full route table readvertisement (route refresh capability). For more information, see RFC 2918, <i>Route Refresh Capability for BGP-4</i> .
Restart time configured on peer	Configured time allowed for restart on the neighbor.
Stale routes from peer are kept for	When graceful restart is negotiated, the maximum time allowed to hold routes from neighbors after the BGP session has gone down.
Peer does not support Restarter functionality	Graceful restart restarter-mode is disabled on the peer.
Peer does not support Receiver functionality	Graceful restart helper-mode is disabled on the peer.
Restart time requested by this peer	Restart time requested by this neighbor during capability negotiation.
Restart flag received from the peer	When this field appears, the BGP speaker has restarted (Restarting), and this peer should not wait for the <b>end-of-rib</b> marker from the speaker before advertising routing information to the speaker.
NLRI that peer supports restart for	Neighbor supports graceful restart for this address family.
NLRI peer can save forwarding state	Neighbor supporting this address family saves all forwarding states.
NLRI that peer saved forwarding for	Neighbor saves all forwarding states for this address family.
NLRI that restart is negotiated for	Router supports graceful restart for this address family.
NLRI of received end-of-rib markers	Address families for which end-of-routing-table markers are received from the neighbor.
NLRI of all end-of-rib markers sent	Address families for which end-of-routing-table markers are sent to the neighbor.
Peer supports 4 byte AS extension (peer-as1)	Peer understands 4-byte AS numbers in BGP messages. The peer is running Junos OS Release 9.1 or later.
NLRIs for which peer can receive multiple paths	Appears in the command output of the local router if the downstream peer is configured to receive multiple BGP routes to a single destination, instead of only receiving the active route.  Possible value is <b>inet-unicast</b> .

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
NLRIs for which peer can send multiple paths: inet-unicast	Appears in the command output of the local router if the upstream peer is configured to send multiple BGP routes to a single destination, instead of only sending the active route.  Possible value is <b>inet-unicast</b> .
Table inet.number	Information about the routing table: <ul style="list-style-type: none"> <li>• <b>RIB State</b>—BGP is in the graceful restart process for this routing table: <b>restart is complete</b> or <b>restart in progress</b>.</li> <li>• <b>Bit</b>—Number that represents the entry in the routing table for this peer.</li> <li>• <b>Send state</b>—State of the BGP group: <b>in sync</b>, <b>not in sync</b>, or <b>not advertising</b>.</li> <li>• <b>Active prefixes</b>—Number of prefixes received from the peer that are active in the routing table.</li> <li>• <b>Received prefixes</b>—Total number of prefixes from the peer, both active and inactive, that are in the routing table.</li> <li>• <b>Accepted prefixes</b>—Total number of prefixes from the peer that have been accepted by a routing policy.</li> <li>• <b>Suppressed due to damping</b>—Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.</li> </ul>
Last traffic (seconds)	Last time any traffic was received from the peer or sent to the peer, and the last time the local routing device checked.
Input messages	Messages that BGP has received from the receive socket buffer, showing the total number of messages, number of update messages, number of times a policy is changed and refreshed, and the buffer size in octets. The buffer size is 16 KB.
Output messages	Messages that BGP has written to the transmit socket buffer, showing the total number of messages, number of update messages, number of times a policy is changed and refreshed, and the buffer size in octets. The buffer size is 16 KB.
Input dropped path attributes	Information about dropped path attributes: <ul style="list-style-type: none"> <li>• <b>Code</b>—Path attribute code.</li> <li>• <b>Count</b>—Path attribute count.</li> </ul>
Input ignored path attributes	Information about ignored path attributes: <ul style="list-style-type: none"> <li>• <b>Code</b>—Path attribute code.</li> <li>• <b>Count</b>—Path attribute count.</li> </ul>
Output queue	Number of BGP packets that are queued to be transmitted to a particular neighbor for a particular routing table. Output queue 0 is for unicast NLRIs, and queue 1 is for multicast NLRIs.  It also specifies the routing table name and the NLRI they represent in the format ( <b>routing table name, NLRI</b> ).  <b>NOTE:</b> The output queues of routing tables that are not advertised, will only show up at <b>extensive</b> output level.
Trace options	Configured tracing of BGP protocol packets and operations.

Table 16: show bgp neighbor Output Fields (*continued*)

Field Name	Field Description
Trace file	Name of the file to receive the output of the tracing operation.
Filter Updates rcv	(orf option only) Number of outbound-route filters received for each configured address family.  <i>NOTE:</i> The counter is cumulative. For example, the counter is increased after the remote peer either resends or clears the outbound route filtering prefix list.
Immediate	(orf option only) Number of route updates received with the immediate flag set. The immediate flag indicates that the BGP peer should readvertise the updated routes.  <i>NOTE:</i> The counter is cumulative. For example, the counter is increased after the remote peer either resends or clears the outbound route filtering prefix list.
Filter	(orf option only) Type of prefix filter received: <b>prefix-based</b> or <b>extended-community</b> .
Received filter entries	(orf option only) List of received filters displayed.
seq	(orf option only) Numerical order assigned to this prefix entry among all the received outbound route filter prefix entries.
prefix	(orf option only) Address for the prefix entry that matches the filter.
minlength	(orf option only) Minimum prefix length, in bits, required to match this prefix.
maxlength	(orf option only) Maximum prefix length, in bits, required to match this prefix.
match	(orf option only) For this prefix match, whether to <b>permit</b> or <b>deny</b> route updates.

## Sample Output

### show bgp neighbor

```

user@host > show bgp neighbor
Peer: 10.255.7.250+179 AS 10   Local: 10.255.7.248+63740 AS 10
  Type: Internal   State: Established   Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ redist_static ]
  Options: <Preference LocalAddress PeerAS Refresh>
  Local Address: 10.255.7.248 Holdtime: 90 Preference: 170 Outbound Timer: 50
  Number of flaps: 0
  Peer ID: 10.255.7.250   Local ID: 10.255.7.248   Active Holdtime: 90
  Keepalive Interval: 30   Group index: 0   Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast

```

```

NLRI of all end-of-rib markers sent: inet-unicast
Peer supports 4 byte AS extension (peer-as 10)
Peer does not support Addpath
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        1
  Accepted prefixes:        1
  Suppressed due to damping: 0
  Advertised prefixes:      1
Last traffic (seconds): Received 9    Sent 5    Checked 5
Input messages: Total 36    Updates 2    Refreshes 0    Octets 718
Output messages: Total 37    Updates 1    Refreshes 0    Octets 796
Output Queue[0]: 0 (inet.0, inet-unicast)

Peer: 10.255.162.214+52193 AS 100 Local: 10.255.167.205+179 AS 100
  Type: Internal    State: Established (route reflector client)Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress Cluster AddressFamily Rib-group Refresh>
  Address families configured: inet-unicast inet-vpn-unicast route-target
  Local Address: 10.255.167.205 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.255.162.214 Local ID: 10.255.167.205 Active Holdtime: 90
  Keepalive Interval: 30 Group index: 0 Peer index: 1

```

### show bgp neighbor (CLNS)

```

user@host> show bgp neighbor
Peer: 10.245.245.1+179 AS 200 Local: 10.245.245.3+3770 AS 100
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Options: <Multihop Preference LocalAddress HoldTime AddressFamily PeerAS
Rib-group Refresh>
  Address families configured: iso-vpn-unicast
  Local Address: 10.245.245.3 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.245.245.1 Local ID: 10.245.245.3 Active Holdtime: 90
  Keepalive Interval: 30 Peer index: 0
  NLRI advertised by peer: iso-vpn-unicast
  NLRI for this session: iso-vpn-unicast
  Peer supports Refresh capability (2)
  Table bgp.isovpn.0 Bit: 10000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:          3
    Received prefixes:        3
    Suppressed due to damping: 0
    Advertised prefixes:      3
  Table aaaa.iso.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:          3
    Received prefixes:        3
    Suppressed due to damping: 0
  Last traffic (seconds): Received 6    Sent 5    Checked 5
  Input messages: Total 1736    Updates 4    Refreshes 0    Octets 33385
  Output messages: Total 1738    Updates 3    Refreshes 0    Octets 33305

```

```
Output Queue[0]: 0 (bgp.isovpn.0, iso-vpn-unicast)
Output Queue[1]: 0 (aaaa.iso.0, iso-vpn-unicast)
```

### show bgp neighbor (Layer 2 VPN)

```
user@host> show bgp neighbor
Peer: 10.69.103.2      AS 65100 Local: 10.69.103.1      AS 65103
  Type: External      State: Active      Flags: <ImportEval>
  Last State: Idle      Last Event: Start
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS
Refresh>
  Address families configured: inet-unicast
  Local Address: 10.69.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.69.104.2      AS 65100 Local: 10.69.104.1      AS 65104
  Type: External      State: Active      Flags: <ImportEval>
  Last State: Idle      Last Event: Start
  Last Error: None
  Export: [ BGP-L-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS
Refresh>
  Address families configured: inet-labeled-unicast
  Local Address: 10.69.104.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.255.14.182+179 AS 69      Local: 10.255.14.176+2131 AS 69
  Type: Internal      State: Established  Flags: <ImportEval>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.255.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.255.14.182      Local ID: 10.255.14.176      Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of received end-of-rib markers: inet-vpn-unicast l2vpn
Table bgp.l3vpn.0 Bit: 10000
  RIB State: BGP restart in progress
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:          10
  Received prefixes:        10
  Suppressed due to damping: 0
Table bgp.l2vpn.0 Bit: 20000
  RIB State: BGP restart in progress
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:          1
```

```

Received prefixes:          1
Suppressed due to damping: 0
Table BGP-INET.inet.0 Bit: 30000
RIB State: BGP restart in progress
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            2
Received prefixes:          2
Suppressed due to damping: 0
Table BGP-L.inet.0 Bit: 40000
RIB State: BGP restart in progress
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            2
Received prefixes:          2
Suppressed due to damping: 0
Table LDP.inet.0 Bit: 50000
RIB State: BGP restart is complete
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            1
Received prefixes:          1
Suppressed due to damping: 0
Table OSPF.inet.0 Bit: 60000
RIB State: BGP restart is complete
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            2
Received prefixes:          2
Suppressed due to damping: 0
Table RIP.inet.0 Bit: 70000
RIB State: BGP restart is complete
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            2
Received prefixes:          2
Suppressed due to damping: 0
Table STATIC.inet.0 Bit: 80000
RIB State: BGP restart is complete
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            1
Received prefixes:          1
Suppressed due to damping: 0
Table L2VPN.l2vpn.0 Bit: 90000
RIB State: BGP restart is complete
RIB State: VPN restart in progress
Send state: in sync
Active prefixes:            1
Received prefixes:          1
Suppressed due to damping: 0
Last traffic (seconds): Received 0    Sent 0    Checked 0
Input messages: Total 14    Updates 13    Refreshes 0    Octets 1053
Output messages: Total 3    Updates 0    Refreshes 0    Octets 105
Output Queue[0]: 0 (bgp.l3vpn.0, inet-vpn-unicast)
Output Queue[1]: 0 (bgp.l2vpn.0, inet-vpn-unicast)
Output Queue[2]: 0 (BGP-INET.inet.0, inet-vpn-unicast)
Output Queue[3]: 0 (BGP-L.inet.0, inet-vpn-unicast)
Output Queue[4]: 0 (LDP.inet.0, inet-vpn-unicast)
Output Queue[5]: 0 (OSPF.inet.0, inet-vpn-unicast)
Output Queue[6]: 0 (RIP.inet.0, inet-vpn-unicast)

```

```
Output Queue[7]: 0 (STATIC.inet.0, inet-vpn-unicast)
Output Queue[8]: 0 (L2VPN.l2vpn.0, inet-vpn-unicast)
```

### show bgp neighbor (Layer 3 VPN)

```
user@host> show bgp neighbor
Peer: 4.4.4.4+179      AS 10045 Local: 5.5.5.5+1214      AS 10045
Type: Internal      State: Established      Flags: <ImportEval>
Last State: OpenConfirm      Last Event: RecvKeepAlive
Last Error: None
Export: [ match-all ] Import: [ match-all ]
Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
      Rib-group Refresh>
Address families configured: inet-vpn-unicast
Local Address: 5.5.5.5 Holdtime: 90 Preference: 170
Flags for NLRI inet-labeled-unicast: TrafficStatistics
Traffic Statistics: Options: all File: /var/log/bstat.log
                        size 131072 files 10

Traffic Statistics Interval: 60
Number of flaps: 0
Peer ID: 192.168.1.110      Local ID: 192.168.1.111      Active Holdtime: 90
Keepalive Interval: 30
NLRI for restart configured on peer: inet-vpn-unicast
NLRI advertised by peer: inet-vpn-unicast
NLRI for this session: inet-vpn-unicast
Peer supports Refresh capability (2)
Restart time configured on the peer: 120
Stale routes from peer are kept for: 300
Restart time requested by this peer: 120
NLRI that peer supports restart for: inet-vpn-unicast
NLRI peer can save forwarding state: inet-vpn-unicast
NLRI that peer saved forwarding for: inet-vpn-unicast
NLRI that restart is negotiated for: inet-vpn-unicast
NLRI of received end-of-rib markers: inet-vpn-unicast
NLRI of all end-of-rib markers sent: inet-vpn-unicast
Table bgp.l3vpn.0 Bit: 10000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          2
  Received prefixes:        2
  Suppressed due to damping: 0
Table vpn-green.inet.0 Bit: 20001
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          2
  Received prefixes:        2
  Suppressed due to damping: 0
Last traffic (seconds): Received 15      Sent 20      Checked 20
Input messages: Total 40      Updates 2      Refreshes 0      Octets 856
Output messages: Total 44      Updates 2      Refreshes 0      Octets 1066
Output Queue[0]: 0 (bgp.l3vpn.0, inet-vpn-unicast)
Output Queue[1]: 0 (vpn-green.inet.0, inet-vpn-unicast)
Trace options: detail packets
Trace file: /var/log/bgpgr.log size 131072 files 10
```

### show bgp neighbor neighbor-address

```
user@host> show bgp neighbor 192.168.1.111
```

```

Peer: 10.255.245.12+179 AS 35 Local: 10.255.245.13+2884 AS 35
  Type: Internal State: Established (route reflector client)Flags: <Sync>
  Last State: OpenConfirm Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime Cluster AddressFamily Rib-group
  Refresh>
  Address families configured: inet-vpn-unicast inet-labeled-unicast
  Local Address: 10.255.245.13 Holdtime: 90 Preference: 170
  Flags for NLRI inet-vpn-unicast: AggregateLabel
  Flags for NLRI inet-labeled-unicast: AggregateLabel
  Number of flaps: 0
  Peer ID: 10.255.245.12 Local ID: 10.255.245.13 Active Holdtime: 90
  Keepalive Interval: 30
BFD: disabled
  NLRI advertised by peer: inet-vpn-unicast inet-labeled-unicast
  NLRI for this session: inet-vpn-unicast inet-labeled-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 300
  Stale routes from peer are kept for: 60
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast inet6-unicast
  NLRI that restart is negotiated for: inet-unicast inet6-unicast
  NLRI of received end-of-rib markers: inet-unicast inet6-unicast
  NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 4
    Received prefixes: 6
    Suppressed due to damping: 0
  Table inet6.0 Bit: 20000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 2
    Suppressed due to damping: 0
  Last traffic (seconds): Received 3 Sent 3 Checked 3
  Input messages: Total 9 Updates 6 Refreshes 0 Octets 403
  Output messages: Total 7 Updates 3 Refreshes 0 Octets 365
  Output Queue[0]: 0 (inet.0, inet-unicast)
  Output Queue[1]: 0 (inet6.0, inet6-unicast)
  Trace options: detail packets
  Trace file: /var/log/bgpr size 131072 files 10

```

### show bgp neighbor neighbor-address

```

user@host> show bgp neighbor 192.168.4.222
Peer: 192.168.4.222+4902 AS 65501 Local: 192.168.4.221+179 AS 65500
  Type: External State: Established Flags: <Sync>
  Last State: OpenConfirm Last Event: RecvKeepAlive
  Last Error: Cease
  Export: [ export-policy ] Import: [ import-policy ]
  Options: <Preference HoldTime AddressFamily PeerAS PrefixLimit Refresh>
  Address families configured: inet-unicast inet-multicast
  Holdtime: 60000 Preference: 170
  Number of flaps: 4
  Last flap event: RecvUpdate
  Error: 'Cease' Sent: 5 Recv: 0
  Peer ID: 10.255.245.6 Local ID: 10.255.245.5 Active Holdtime: 60000
  Keepalive Interval: 20000 Peer index: 0
  BFD: disabled, down

```

```

Local Interface: fxp0.0
NLRI advertised by peer: inet-unicast inet-multicast
NLRI for this session: inet-unicast inet-multicast
Peer supports Refresh capability (2)
Table inet.0 Bit: 10000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:           8
  Received prefixes:        10
  Accepted prefixes:        10
  Suppressed due to damping: 0
  Advertised prefixes:      3
Table inet.2 Bit: 20000
  RIB State: BGP restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Accepted prefixes:         0
  Suppressed due to damping: 0
  Advertised prefixes:       0
Last traffic (seconds): Received 357 Sent 357 Checked 357
Input messages: Total 4 Updates 2 Refreshes 0 Octets 211
Output messages: Total 4 Updates 1 Refreshes 0 Octets 147
Output Queue[0]: 0 (inet.0, inet-unicast)
Output Queue[1]: 0 (inet.2, inet-multicast)
Trace options: all
Trace file: /var/log/bgp size 10485760 files 10

```

#### show bgp neighbor orf neighbor-address detail

```

user@host > show bgp neighbor orf 192.168.165.56 detail
Peer: 192.168.165.56+179 Type: External
Group: ext1

inet-unicast
  Filter updates rcv:           1 Immediate:           1
  Filter: prefix-based receive
  Received filter entries:
    seq 1: prefix 2.2.2.2/32: minlen 32: maxlen 32: match deny:

inet6-unicast
  Filter updates rcv:           0 Immediate:           1
  Filter: prefix-based receive
  Received filter entries:
    *.*

```

## show bgp replication

<b>Syntax</b>	<b>show bgp replication</b>
<b>Release Information</b>	Command introduced in JUNOS Release 8.5. Command introduced in Junos OS Release 11.3 for the QFX Series. Support for <b>logical-system</b> option introduced in Junos OS Release 13.3
<b>Description</b>	Displays the status of BGP state replication between the master and backup Routing Engines on devices that have nonstop active routing configured on them.
<b>Options</b>	<b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><a href="#">show bgp replication logical-system on page 865</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show bgp replication (for Master) on page 863</a> <a href="#">show bgp replication (for Backup) on page 863</a>
<b>Output Fields</b>	<a href="#">Table 17 on page 862</a> lists the output fields for the <b>show bgp replication</b> command. Output fields are listed in the approximate order in which they appear.

**Table 17: show bgp replication Output Fields**

Field Name	Field Description
<b>session state</b>	State of the current internal BGP state replication session, Up or Down, and the duration for which the session has been in the indicated state.
<b>flaps</b>	Total number of flaps that occurred.
<b>protocol state</b>	Current state of the protocol operation, Active, Connect, Idle, and the duration for which the protocol has been in the indicated state.
<b>synchronization state</b>	Synchronization state at the time of executing the command. The states can be: <ul style="list-style-type: none"> <li><b>Idle</b></li> <li><b>Neighbor</b>—Indicates that the neighbor state synchronization is in progress.</li> <li><b>AckWait</b>—Indicates that the request processing is over.</li> <li><b>ORF</b>—Indicates that the outbound routing filter synchronization is in progress.</li> <li><b>RIB</b>—Indicates that the routing table synchronization is in progress.</li> <li><b>Complete</b></li> </ul>

Table 17: show bgp replication Output Fields (*continued*)

Field Name	Field Description
<b>number of peers waiting</b>	<p>Total number of peers waiting for various messages:</p> <ul style="list-style-type: none"> <li>• <b>AckWait</b>—Number of peers waiting for a connection establishment or completed acknowledgment messages.</li> <li>• <b>SoWait</b>—Number of peers waiting for TCP socket-related operations.</li> <li>• <b>Scheduled</b>—Number of peers being synchronized.</li> </ul>
<b>messages sent</b>	<p>Number of various types of messages that have been sent since internal replication session became active:</p> <ul style="list-style-type: none"> <li>• <b>Open</b>—Number of Open messages sent.</li> <li>• <b>Establish</b>—Number of connection establishment acknowledgment messages sent.</li> <li>• <b>Update</b>—Number of update messages sent.</li> <li>• <b>Error</b>—Number of error messages sent.</li> <li>• <b>Complete</b>—Number of connection complete acknowledgment messages sent.</li> </ul>
<b>messages received</b>	<p>Total number of messages received:</p> <ul style="list-style-type: none"> <li>• <b>Open</b>—Number of Open messages received.</li> <li>• <b>Request</b>—Number of request messages received: <ul style="list-style-type: none"> <li>• <b>Wildcard</b>—Number of requests received that used wildcards in the target address.</li> <li>• <b>Targeted</b>—Number of requests received that used a specific address.</li> </ul> </li> <li>• <b>EstablishAck</b>—Number of connection establishment acknowledgement messages received.</li> <li>• <b>CompleteAck</b>—Number of connection completed acknowledgement messages received.</li> </ul>

## Sample Output

### show bgp replication (for Master)

```

user@host> show bgp replication
Synchronization master:
  Session state: Up, Since: 44:07
  Flaps: 0
  Protocol state: Idle, Since: 14
  Synchronization state: Complete
  Number of peers waiting: AckWait: 0, SoWait: 0, Scheduled: 0
  Messages sent: Open 1, Establish 924, Update 381, Error 60, Complete 114
  Messages received: Open 1, Request 1 wildcard 113 targeted, EstablishAck 924,
  CompleteAck 114

```

### show bgp replication (for Backup)

```

user@host> show bgp replication
Synchronization backup:
  State: Established 13 ago
  , Unsync timer: 2

  Unsync entry queue:
    Instance: 0 Neighbor: 30.30.30.1 elapsed: 7
    Instance: 0 Neighbor: 40.40.40.3 elapsed: 7
    Instance: 0 Neighbor: 40.40.40.4 elapsed: 7

```

```
Instance: 0 Neighbor: 40.40.40.5 elapsed: 7  
Instance: 0 Neighbor: 40.40.40.6 elapsed: 7  
Instance: 0 Neighbor: 40.40.40.1 elapsed: 7  
Instance: 0 Neighbor: 40.40.40.2 elapsed: 7
```

## show bgp replication logical-system

<b>Syntax</b>	show bgp replication logical-system <logical-system-name>
<b>Release Information</b>	Command introduced in Junos OS Release 13.3.
<b>Description</b>	Display logical system-specific BGP state replication between the master and backup logical system on Routing Engines that have nonstop active routing configured on them.
<b>Options</b>	This command has no options.
<b>Required Privilege Level</b>	View
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show bgp replication on page 862</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show bgp replication logical-system on page 866</a>
<b>Output Fields</b>	Table 18 on page 865 lists the output fields for the <b>show bgp replication logical-system</b> command. Output fields are listed in the approximate order in which they appear.

**Table 18: show bgp replication logical-system Output Fields**

Field Name	Field Description
<b>session state</b>	State of the current internal BGP state replication session, Up or Down, and the duration for which the session has been in the indicated state.
<b>flaps</b>	Total number of flaps that occurred.
<b>protocol state</b>	Current state of the protocol operation (Active, Connect, Idle) and the duration for which the protocol has been in the indicated state.
<b>synchronization state</b>	<p>Synchronization state at the time of executing the command. The states can be:</p> <ul style="list-style-type: none"> <li>• <b>Idle</b></li> <li>• <b>Neighbor</b>—Indicates that the neighbor state synchronization is in progress.</li> <li>• <b>AckWait</b>—Indicates that the request processing is over.</li> <li>• <b>ORF</b>—Indicates that the outbound routing filter synchronization is in progress.</li> <li>• <b>RIB</b>—Indicates that the routing table synchronization is in progress.</li> <li>• <b>Complete</b></li> </ul>
<b>number of peers waiting</b>	<p>Total number of peers waiting for various messages:</p> <ul style="list-style-type: none"> <li>• <b>AckWait</b>—Number of peers waiting for connection establishment or completed acknowledgment messages.</li> <li>• <b>SoWait</b>—Number of peers waiting for TCP socket-related operations.</li> <li>• <b>Scheduled</b>—Number of peers being synchronized.</li> </ul>

Table 18: show bgp replication logical-system Output Fields (*continued*)

Field Name	Field Description
<b>messages sent</b>	<p>Number of various types of messages that have been sent since internal replication session became active:</p> <ul style="list-style-type: none"> <li>• <b>Open</b>—Number of Open messages sent.</li> <li>• <b>Establish</b>—Number of connection establishment acknowledgment messages sent.</li> <li>• <b>Update</b>—Number of update messages sent.</li> <li>• <b>Error</b>—Number of error messages sent.</li> <li>• <b>Complete</b>—Number of connection complete acknowledgment messages sent.</li> </ul>
<b>messages received</b>	<p>Total number of messages received:</p> <ul style="list-style-type: none"> <li>• <b>Open</b>—Number of Open messages received.</li> <li>• <b>Request</b>—Number of request messages received: <ul style="list-style-type: none"> <li>• <b>Wildcard</b>—Number of requests received that used wildcards in the target address.</li> <li>• <b>Targeted</b>—Number of requests received that used a specific address.</li> </ul> </li> <li>• <b>EstablishAck</b>—Number of connection establishment acknowledged messages received.</li> <li>• <b>CompleteAck</b>—Number of connection completed acknowledged messages received.</li> </ul>

## Sample Output

### show bgp replication logical-system

```

user@host> show bgp replication logical-system lr2
Synchronization master:
  Session state: Up, Since: 24:53
  Flaps: 0
  Protocol state: Idle, Since: 2
  Synchronization state: Complete
  Number of peers waiting: AckWait: 0, SoWait: 0, Scheduled: 0
  Messages sent: Open 1, Establish 145, Update 0, Error 1, Complete 145
  Messages received: Open 1, Request 1 wildcard 144 targeted, EstablishAck 0,
  CompleteAck 145

```

## show bgp summary

<b>List of Syntax</b>	<a href="#">Syntax on page 867</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 867</a>
<b>Syntax</b>	<pre>show bgp summary &lt;exact-instance <i>instance-name</i>&gt; &lt;group <i>group-name</i>&gt; &lt;instance <i>instance-name</i>&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	<pre>show bgp summary &lt;exact-instance <i>instance-name</i>&gt; &lt;instance <i>instance-name</i>&gt;</pre>
<b>Release Information</b>	<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 11.3 for the QFX Series.</p> <p><b>exact-instance</b> option introduced in Junos OS Release 11.4.</p> <p><b>group</b> option introduced in Junos OS Release 13.3.</p>
<b>Description</b>	Display BGP summary information.
<b>Options</b>	<p><b>none</b>—Display BGP summary information for all routing instances.</p> <p><b>exact-instance <i>instance-name</i></b>—(Optional) Display information for the specified instance only.</p> <p><b>group</b>—Display overview of bgp information for a particular group</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information for all routing instances whose name begins with this string (for example, <b>cust1</b>, <b>cust11</b>, and <b>cust111</b> are all displayed when you run the <b>show bgp summary instance cust1</b> command). The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show bgp summary (When a Peer Is Not Established) on page 870</a> <a href="#">show bgp summary (When a Peer Is Established) on page 870</a> <a href="#">show bgp summary (CLNS) on page 870</a> <a href="#">show bgp summary (Layer 2 VPN) on page 871</a> <a href="#">show bgp summary (Layer 3 VPN) on page 871</a> <a href="#">show bgp summary group on page 871</a>
<b>Output Fields</b>	<p><a href="#">Table 19 on page 868</a> describes the output fields for the <b>show bgp summary</b> command. Output fields are listed in the approximate order in which they appear.</p>

Table 19: show bgp summary Output Fields

Field Name	Field Description
<b>Groups</b>	Number of BGP groups.
<b>Peers</b>	Number of BGP peers.
<b>Down peers</b>	Number of down BGP peers.
<b>Table</b>	Name of routing table.
<b>Tot Paths</b>	Total number of paths.
<b>Act Paths</b>	Number of active routes.
<b>Suppressed</b>	Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.
<b>History</b>	Number of withdrawn routes stored locally to keep track of damping history.
<b>Damp State</b>	Number of routes with a figure of merit greater than zero, but still active because the value has not reached the threshold at which suppression occurs.
<b>Pending</b>	Routes in process by BGP import policy.
<b>Peer</b>	Address of each BGP peer. Each peer has one line of output.
<b>AS</b>	Peer's AS number.
<b>InPkt</b>	Number of packets received from the peer.
<b>OutPkt</b>	Number of packets sent to the peer.
<b>OutQ</b>	Number of BGP packets that are queued to be transmitted to a particular neighbor. It normally is 0 because the queue usually is emptied quickly.
<b>Flaps</b>	Number of times the BGP session has gone down and then come back up.
<b>Last Up/Down</b>	Last time since the neighbor transitioned to or from the established state.

Table 19: show bgp summary Output Fields (*continued*)

Field Name	Field Description
<b>State #Active /Received/Accepted /Damped</b>	<p>Multipurpose field that displays information about BGP peer sessions. The field's contents depend upon whether a session is established and whether it was established on the main routing device or in a routing instance.</p> <ul style="list-style-type: none"> <li>If a peer is not established, the field shows the state of the peer session: <b>Active</b>, <b>Connect</b>, or <b>Idle</b>. In general, the Idle state is the first stage of a connection. BGP is waiting for a Start event. A session can be idle for other reasons as well. The reason that a session is idle is sometimes displayed. For example: <b>Idle (Removal in progress)</b> or <b>Idle (LicenseFailure)</b>.</li> <li>If a BGP session is established on the main routing device, the field shows the number of active, received, accepted, and damped routes that are received from a neighbor and appear in the <b>inet.0</b> (main) and <b>inet.2</b> (multicast) routing tables. For example, <b>8/10/10/2</b> and <b>2/4/4/0</b> indicate the following: <ul style="list-style-type: none"> <li>8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the <b>inet.0</b> routing table.</li> <li>2 active routes, 4 received routes, 4 accepted routes, and no damped routes from a BGP peer appear in the <b>inet.2</b> routing table.</li> </ul> </li> <li>If a BGP session is established in a routing instance, the field indicates the established (<b>Establ</b>) state, identifies the specific routing table that receives BGP updates, and shows the number of active, received, and damped routes that are received from a neighbor. For example, <b>Establ VPN-AB.inet.0: 2/4/0</b> indicates the following: <ul style="list-style-type: none"> <li>The BGP session is established.</li> <li>Routes are received in the <b>VPN-AB.inet.0</b> routing table.</li> <li>The local routing device has two active routes, four received routes, and no damped routes from a BGP peer.</li> </ul> </li> </ul> <p>When a BGP session is established, the peers are exchanging update messages.</p>

## Sample Output

### show bgp summary (When a Peer Is Not Established)

```

user@host> show bgp summary
Groups: 2 Peers: 4 Down peers: 1
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 6 4 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.3 65002 86 90 0 2 42:54 0/0/0

0/0/0
10.0.0.4 65002 90 91 0 1 42:54 0/2/0

0/0/0
10.0.0.6 65002 87 90 0 3 3 Active
10.1.12.1 65001 89 89 0 1 42:54 4/4/0

0/0/0

```

### show bgp summary (When a Peer Is Established)

```

user@host> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 6 4 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2 65002 88675 88652 0 2 42:38 2/4/0

0/0/0
10.0.0.3 65002 54528 54532 0 1 2w4d22h 0/0/0

0/0/0
10.0.0.4 65002 51597 51584 0 0 2w3d22h 2/2/0

0/0/0

user@host> show bgp summary logical-system R3
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.13vpn.0 2 2 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
1.1.1.2 2 204 206 0 0 1:30:59
Establ
bgp.13vpn.0: 2/2/2/0
red.inet.0: 2/2/2/0
10.1.1.10 3 206 207 0 0 1:31:36
Establ
red.inet.0: 2/2/2/0

```

### show bgp summary (CLNS)

```

user@host> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.245.245.1 200 1735 1737 0 0 14:26:12 Establ

```

```

bgp.isovpn.0: 3/3/0
aaaa.iso.0: 3/3/0

```

### show bgp summary (Layer 2 VPN)

```

user@host> show bgp summary
Groups: 1 Peers: 5 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l2vpn.0 1 1 0 0 0
inet.0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.245.35 65299 72 74 0 1 19:00 Establ
bgp.l2vpn.0: 1/1/0
frame-vpn.l2vpn.0: 1/1/0
10.255.245.36 65299 2164 2423 0 4 19:50 Establ
bgp.l2vpn.0: 0/0/0
frame-vpn.l2vpn.0: 0/0/0
10.255.245.37 65299 36 37 0 4 17:07 Establ
inet.0: 0/0/0
10.255.245.39 65299 138 168 0 6 53:48 Establ
bgp.l2vpn.0: 0/0/0
frame-vpn.l2vpn.0: 0/0/0
10.255.245.69 65299 134 140 0 6 53:42 Establ
inet.0: 0/0/0

```

### show bgp summary (Layer 3 VPN)

```

user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 2 2 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.39.1.5 2 21 22 0 0 6:26 Establ
VPN-AB.inet.0: 1/1/0
10.255.71.15 1 19 21 0 0 6:17 Establ
bgp.l3vpn.0: 2/2/0
VPN-A.inet.0: 1/1/0
VPN-AB.inet.0: 2/2/0
VPN-B.inet.0: 1/1/0

```

### show bgp summary group

```

user@host> show bgp summary group Group2
Groups: 3 Peers: 3 Down peers: 3
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.1 56 0 0 0 0 51
Idle

user@host> show bgp summary logical-system R3 group toR4
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 2 2 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.1.1.10 3 207 207 0 0 1:31:40

```

```
Estab1  
red.inet.0: 2/2/2/0
```

## Sample Output

## show policy damping

<b>List of Syntax</b>	<a href="#">Syntax on page 873</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 873</a>
<b>Syntax</b>	<pre>show policy damping &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	show policy damping
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Display information about BGP route flap damping parameters.
<b>Options</b>	<p><b>none</b>—Display information about BGP route flap damping parameters.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Additional Information</b>	In the output from this command, figure-of-merit values correlate with the probability of future instability of a routing device. Routes with higher figure-of-merit values are suppressed for longer periods of time. The figure-of-merit value decays exponentially over time. A figure-of-merit value of zero is assigned to each new route. The value is increased each time the route is withdrawn or readvertised, or when one of its path attributes changes.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• “Configuring BGP Flap Damping Parameters” in the <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i></li> <li>• <a href="#">clear bgp damping on page 799</a></li> <li>• <a href="#">show route damping on page 910</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show policy damping on page 874</a>
<b>Output Fields</b>	<p><a href="#">Table 20 on page 874</a> describes the output fields for the <b>show policy damping</b> command. Output fields are listed in the approximate order in which they appear.</p>

Table 20: show policy damping Output Fields

Field Name	Field Description
<b>Halflife</b>	Decay half-life, in minutes. The value represents the period during which the accumulated figure-of-merit value is reduced by half if the route remains stable. If a route has flapped, but then becomes stable, the figure-of-merit value for the route decays exponentially. For example, for a route with a figure-of-merit value of 1500, if no incidents occur, its figure-of-merit value is reduced to 750 after 15 minutes and to 375 after another 15 minutes.
<b>Reuse merit</b>	Figure-of-merit value below which a suppressed route can be used again. A suppressed route becomes reusable when its figure-of-merit value decays to a value below a reuse threshold, and the route once again is considered usable and can be installed in the forwarding table and exported from the routing table.
<b>Suppress/cutoff merit</b>	Figure-of-merit value above which a route is suppressed for use or inclusion in advertisements. When a route's figure-of-merit value reaches a particular level, called the cutoff or suppression threshold, the route is suppressed. When a route is suppressed, the routing table no longer installs the route into the forwarding table and no longer exports this route to any of the routing protocols.
<b>Maximum suppress time</b>	Maximum hold-down time, in minutes. The value represents the maximum time that a route can be suppressed no matter how unstable it has been before this period of stability.
<b>Computed values</b>	<ul style="list-style-type: none"> <li>• <b>Merit ceiling</b>—Maximum merit that a flapping route can collect.</li> <li>• <b>Maximum decay</b>—Maximum decay half-life, in minutes.</li> </ul>

## Sample Output

### show policy damping

```

user@host> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "standard-damping":
  Halflife: 10 minutes
  Reuse merit: 4000 Suppress/cutoff merit: 8000
  Maximum suppress time: 30 minutes
  Computed values:
    Merit ceiling: 32120
    Maximum decay: 12453

```

## show policy

<b>List of Syntax</b>	<a href="#">Syntax on page 875</a> <a href="#">Syntax (EX Series Switches) on page 875</a>
<b>Syntax</b>	<pre>show policy &lt;logical-system (all   <i>logical-system-name</i>)&gt; &lt;<i>policy-name</i>&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show policy &lt;<i>policy-name</i>&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display information about configured routing policies.
<b>Options</b>	<p><b>none</b>—List the names of all configured routing policies.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b><i>policy-name</i></b>—(Optional) Show the contents of the specified policy.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show policy damping on page 873</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show policy on page 876</a> <a href="#">show policy <i>policy-name</i> on page 876</a> <a href="#">show policy (Multicast Scoping) on page 876</a>
<b>Output Fields</b>	<p><a href="#">Table 21 on page 875</a> lists the output fields for the <b>show policy</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 21: 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

<b>Syntax</b>	<pre>show policy conditions &lt;condition-name&gt; &lt;detail&gt; &lt;dynamic&gt; &lt;logical-system (all   logical-system-name)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show policy conditions &lt;condition-name&gt; &lt;detail&gt; &lt;dynamic&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.0.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	<p>Display all the configured conditions as well as the routing tables with which the configuration manager is interacting. If the <b>detail</b> keyword is included, the output also displays dependent routes for each condition.</p>
<b>Options</b>	<p><b>none</b>—Display all configured conditions and associated routing tables.</p> <p><b>condition-name</b>—(Optional) Display information about the specified condition only.</p> <p><b>detail</b>—(Optional) Display the specified level of output.</p> <p><b>dynamic</b>—(Optional) Display information about the conditions in the dynamic database.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show policy conditions detail on page 878</a>
<b>Output Fields</b>	<p><a href="#">Table 22 on page 877</a> lists the output fields for the <b>show policy conditions</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 22: show policy conditions Output Fields**

Field Name	Field Description	Level of Output
<b>Condition</b>	Name of configured condition.	All levels
<b>event</b>	Condition type. If the <b>if-route-exists</b> option is configured, the event type is: <b>Existence of a route in a specific routing table.</b>	All levels
<b>Dependent routes</b>	List of routes dependent on the condition, along with the latest generation number.	<b>detail</b>
<b>Condition tables</b>	List of routing tables associated with the condition, along with the latest generation number and number of dependencies.	All levels

Table 22: 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 primary (static), event: Existence of a route in a specific routing
table
Dependent routes:
  8.41.0.0/24, generation 18

Condition standby (static), event: Existence of a route in a specific routing
table
Dependent routes:
  8.41.0.0/24, generation 18

Condition tables:
Table mpls.0, generation 0, dependencies 0, If-route-exists conditions: primary
(static) standby (static)
Table l3vpn.inet.0, generation 633, dependencies 2
```

## show policy damping

<b>List of Syntax</b>	<a href="#">Syntax on page 879</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 879</a>
<b>Syntax</b>	<pre>show policy damping &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switch and QFX Series)</b>	show policy damping
<b>Release Information</b>	<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 11.3 for the QFX Series.</p>
<b>Description</b>	Display information about BGP route flap damping parameters.
<b>Options</b>	<p><b>none</b>—Display information about BGP route flap damping parameters.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Additional Information</b>	In the output from this command, figure-of-merit values correlate with the probability of future instability of a routing device. Routes with higher figure-of-merit values are suppressed for longer periods of time. The figure-of-merit value decays exponentially over time. A figure-of-merit value of zero is assigned to each new route. The value is increased each time the route is withdrawn or readvertised, or when one of its path attributes changes.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• “Configuring BGP Flap Damping Parameters” in the <i>Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices</i></li> <li>• <a href="#">clear bgp damping on page 799</a></li> <li>• <a href="#">show route damping on page 910</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show policy damping on page 880</a>
<b>Output Fields</b>	<p><a href="#">Table 20 on page 874</a> describes the output fields for the <b>show policy damping</b> command. Output fields are listed in the approximate order in which they appear.</p>

Table 23: show policy damping Output Fields

Field Name	Field Description
<b>Halflife</b>	Decay half-life, in minutes. The value represents the period during which the accumulated figure-of-merit value is reduced by half if the route remains stable. If a route has flapped, but then becomes stable, the figure-of-merit value for the route decays exponentially. For example, for a route with a figure-of-merit value of 1500, if no incidents occur, its figure-of-merit value is reduced to 750 after 15 minutes and to 375 after another 15 minutes.
<b>Reuse merit</b>	Figure-of-merit value below which a suppressed route can be used again. A suppressed route becomes reusable when its figure-of-merit value decays to a value below a reuse threshold, and the route once again is considered usable and can be installed in the forwarding table and exported from the routing table.
<b>Suppress/cutoff merit</b>	Figure-of-merit value above which a route is suppressed for use or inclusion in advertisements. When a route's figure-of-merit value reaches a particular level, called the cutoff or suppression threshold, the route is suppressed. When a route is suppressed, the routing table no longer installs the route into the forwarding table and no longer exports this route to any of the routing protocols.
<b>Maximum suppress time</b>	Maximum hold-down time, in minutes. The value represents the maximum time that a route can be suppressed no matter how unstable it has been before this period of stability.
<b>Computed values</b>	<ul style="list-style-type: none"> <li>• <b>Merit ceiling</b>—Maximum merit that a flapping route can collect.</li> <li>• <b>Maximum decay</b>—Maximum decay half-life, in minutes.</li> </ul>

## Sample Output

### show policy damping

```

user@host> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "standard-damping":
  Halflife: 10 minutes
  Reuse merit: 4000 Suppress/cutoff merit: 8000
  Maximum suppress time: 30 minutes
  Computed values:
    Merit ceiling: 32120
    Maximum decay: 12453

```

## show route

<b>List of Syntax</b>	<a href="#">Syntax on page 881</a> <a href="#">Syntax (EX Series Switches) on page 881</a>
<b>Syntax</b>	<pre>show route &lt;all&gt; &lt;destination-prefix&gt; &lt;logical-system (all   logical-system-name)&gt; &lt;private&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route &lt;all&gt; &lt;destination-prefix&gt; &lt;private&gt;</pre>
<b>Release Information</b>	<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 <b>private</b> introduced in Junos OS Release 9.5.</p> <p>Option <b>private</b> introduced in Junos OS Release 9.5 for EX Series switches.</p>
<b>Description</b>	Display the active entries in the routing tables.
<b>Options</b>	<p><b>none</b>—Display brief information about all active entries in the routing tables.</p> <p><b>all</b>—(Optional) Display information about all routing tables, including private, or internal, routing tables.</p> <p><b>destination-prefix</b>—(Optional) Display active entries for the specified address or range of addresses.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>private</b>—(Optional) Display information only about all private, or internal, routing tables.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Configuring RIP</i></li> <li>• <i>Example: Configuring RIPng</i></li> <li>• <i>Example: Configuring IS-IS</i></li> <li>• <a href="#">Examples: Configuring Internal BGP Peering on page 56</a></li> <li>• <a href="#">Examples: Configuring External BGP Peering on page 33</a></li> <li>• <i>Examples: Configuring OSPF Routing Policy</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show route on page 884</a> <a href="#">show route on page 885</a>

[show route destination-prefix on page 885](#)

[show route extensive on page 885](#)

**Output Fields** [Table 24 on page 882](#) describes the output fields for the **show route** command. Output fields are listed in the approximate order in which they appear.

**Table 24: 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"> <li>• <b>active</b> (routes that are active).</li> <li>• <b>holddown</b> (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.</li> </ul> <p>However, if you have configured advertisement of multiple routes (with the <a href="#">add-path</a> or <a href="#">advertise-inactive</a> 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"> <li>• <b>hidden</b> (routes that are not used because of a routing policy).</li> </ul>
<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"> <li>• <b>MPLS-label</b> (for example, 80001).</li> <li>• <b>interface-name</b> (for example, ge-1/0/2).</li> <li>• <b>neighbor-address:control-word-status:encapsulation type:vc-id:source</b> (Layer 2 circuit only. For example, 10.1.1.195:NoCtrlWord:1:1:Local/96): <ul style="list-style-type: none"> <li>• <b>neighbor-address</b>—Address of the neighbor.</li> <li>• <b>control-word-status</b>—Whether the use of the control word has been negotiated for this virtual circuit: <b>NoCtrlWord</b> or <b>CtrlWord</b>.</li> <li>• <b>encapsulation type</b>—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.</li> <li>• <b>vc-id</b>—Virtual circuit identifier.</li> <li>• <b>source</b>—Source of the advertisement: <b>Local</b> or <b>Remote</b>.</li> </ul> </li> </ul>

Table 24: show route Output Fields (*continued*)

Field Name	Field Description
[ <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"> <li>• +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table.</li> <li>• - —A hyphen indicates the last active route.</li> <li>• *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a <b>to</b> line indicates the best subpath to the route.</li> </ul> <p>In every routing metric except for the BGP <b>LocalPref</b> attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the <b>LocalPref</b> value in the <b>Preference2</b> field. For example, if the <b>LocalPref</b> value for Route 1 is 100, the <b>Preference2</b> value is -101. If the <b>LocalPref</b> value for Route 2 is 155, the <b>Preference2</b> value is -156. Route 2 is preferred because it has a higher <b>LocalPref</b> value and a lower <b>Preference2</b> value.</p>
<i>weeks:days</i> <i>hours:minutes:seconds</i>	How long the route been known (for example, <b>2w4d 13:11:14</b> , 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"> <li>• I—IGP.</li> <li>• E—EGP.</li> <li>• ?—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• [ ]—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.</li> <li>• { }—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.</li> <li>• ( )—Parentheses enclose a confederation.</li> <li>• ( [ ] )—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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>

Table 24: show route Output Fields (*continued*)

Field Name	Field Description
<b>validation-state</b>	<p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> <li>• <b>Invalid</b>—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.</li> <li>• <b>Unknown</b>—Indicates that the prefix is not among the prefixes or prefix ranges in the database.</li> <li>• <b>Unverified</b>—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers.</li> <li>• <b>Valid</b>—Indicates that the prefix and autonomous system pair are found in the database.</li> </ul>
<b>to</b>	<p>Next hop to the destination. An angle bracket (&gt;) indicates that the route is the selected route.</p> <p>If the destination is <b>Discard</b>, traffic is dropped.</p>
<b>via</b>	<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 <b>Selected</b>. This field can also contain the following information:</p> <ul style="list-style-type: none"> <li>• <b>Weight</b>—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.</li> <li>• <b>Balance</b>—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.</li> <li>• <b>lsp-path-name</b>—Name of the LSP used to reach the next hop.</li> <li>• <b>label-action</b>—MPLS label and operation occurring at the next hop. The operation can be <b>pop</b> (where a label is removed from the top of the stack), <b>push</b> (where another label is added to the label stack), or <b>swap</b> (where a label is replaced by another label). For VPNs, expect to see multiple <b>push</b> operations, corresponding to the inner and outer labels required for VPN routes (in the case of a direct PE-to-PE connection, the VPN route would have the inner label push only).</li> </ul>

## 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 1t-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 1t-0/3/0.28, label-switched-path toF

```

## show route

The following sample output shows a VPN route with composite next hops enabled. The first **Push** operation corresponds to the outer label. The second **Push** operation corresponds to the inner label.

```
user@host> show route 70.0.0.0
```

```

13979:665001.inet.0: 871 destinations, 3556 routes (871 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

```

```

70.0.0.0/24      @[BGP/170] 00:28:32, localpref 100, from 10.9.9.160
                  AS path: 13980 ?, validation-state: unverified
                  > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                  [BGP/170] 00:28:28, localpref 100, from 10.9.9.169
                  AS path: 13980 ?, validation-state: unverified
                  > to 10.100.0.42 via ae2.0, Push 126016, Push 300368(top)
                  #[Multipath/255] 00:28:28, metric2 102
                  > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                  to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)

```

## 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
        Validation State: unverified
        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

<b>List of Syntax</b>	<a href="#">Syntax on page 887</a> <a href="#">Syntax (EX Series Switches) on page 887</a>
<b>Syntax</b>	<pre>show route active-path &lt;brief   detail   extensive   terse&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route active-path &lt;brief   detail   extensive   terse&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 8.0.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display all active routes for destinations. An active route is a route that is selected as the best path. Inactive routes are not displayed.
<b>Options</b>	<p><b>none</b>—Display all active routes.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b>.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route active-path on page 887</a> <a href="#">show route active-path brief on page 888</a> <a href="#">show route active-path detail on page 888</a> <a href="#">show route active-path extensive on page 889</a> <a href="#">show route active-path terse on page 891</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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 887](#).

### 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-kernel 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			>1o0.0	
*	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

<b>Syntax</b>	<code>show route advertising-protocol <i>protocol</i> <i>neighbor-address</i></code> <code>&lt;brief   detail   extensive   terse&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4.
<b>Description</b>	Display the routing information as it has been prepared for advertisement to a particular neighbor of a particular dynamic routing protocol.
<b>Options</b>	<p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b><i>neighbor-address</i></b>—Address of the neighboring router to which the route entry is being transmitted.</p> <p><b><i>protocol</i></b>—Protocol transmitting the route:</p> <ul style="list-style-type: none"> <li>• <b>bgp</b>—Border Gateway Protocol</li> <li>• <b>dvmrp</b>—Distance Vector Multicast Routing Protocol</li> <li>• <b>msdp</b>—Multicast Source Discovery Protocol</li> <li>• <b>pim</b>—Protocol Independent Multicast</li> <li>• <b>rip</b>—Routing Information Protocol</li> <li>• <b>ripng</b>—Routing Information Protocol next generation</li> </ul>
<b>Additional Information</b>	Routes displayed are routes that the routing table has exported into the routing protocol and that have been filtered by the associated protocol's <b>export</b> routing policy statements.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring the MED Attribute Directly on page 95</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show route advertising-protocol bgp (Layer 3 VPN) on page 894</a> <a href="#">show route advertising-protocol bgp detail on page 895</a> <a href="#">show route advertising-protocol bgp detail (Layer 2 VPN) on page 895</a> <a href="#">show route advertising-protocol bgp detail (Layer 3 VPN) on page 895</a> <a href="#">show route advertising-protocol bgp extensive all (Next Hop Self with RIB-out IP Address) on page 895</a>
<b>Output Fields</b>	<a href="#">Table 25 on page 893</a> lists the output fields for the <b>show route advertising-protocol</b> command. Output fields are listed in the approximate order in which they appear.

Table 25: 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"> <li>• <b>active</b> (routes that are active)</li> <li>• <b>holddown</b> (routes that are in the pending state before being declared inactive)</li> <li>• <b>hidden</b> (routes that are not used because of a routing policy)</li> </ul>	All levels
<b>Prefix</b>	Destination prefix.	<b>brief none</b>
<i>destination-prefix (entry, announced)</i>	Destination prefix. The <b>entry</b> value is the number of routes for this destination, and the <b>announced</b> value is the number of routes being announced for this destination.	<b>detail extensive</b>
<b>BGP group and type</b>	BGP group name and type ( <b>Internal</b> or <b>External</b> ).	<b>detail extensive</b>
<b>Route Distinguisher</b>	Unique 64-bit prefix augmenting each IP subnet.	<b>detail extensive</b>
<b>Advertised Label</b>	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.	<b>detail extensive</b>
<b>Label-Base, range</b>	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.	<b>detail extensive</b>
<b>VPN Label</b>	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.	<b>detail extensive</b>
<b>Nexthop</b>	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 <b>Self</b> , and the RIB-out next hop is a specific IP address, the RIB-out IP address is included in the extensive output. See <a href="#">show route advertising-protocol bgp extensive all (Next Hop Self with RIB-out IP Address)</a> on page 895.	All levels
<b>MED</b>	Multiple exit discriminator value included in the route.	<b>brief</b>
<b>Lclpref or Localpref</b>	Local preference value included in the route.	All levels

Table 25: show route advertising-protocol Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>AS path</b>	<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"> <li>• <b>I</b>—IGP.</li> <li>• <b>E</b>—EGP.</li> <li>• <b>?</b>—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• <b>[ ]</b>—Brackets enclose the local AS number associated with the AS path if configured on the router, or if AS path prepending is configured.</li> <li>• <b>{ }</b>—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.</li> <li>• <b>( )</b>—Parentheses enclose a confederation.</li> <li>• <b>( [ ] )</b>—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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
<b>Communities</b>	Community path attribute for the route. See the output field table for the <a href="#">show route detail</a> command for all possible values for this field.	<b>detail extensive</b>
<b>AIGP</b>	Accumulated interior gateway protocol (AIGP) BGP attribute.	<b>detail extensive</b>
<b>Attrset AS</b>	Number, local preference, and path of the autonomous system (AS) that originated the route. These values are stored in the <b>Attrset</b> attribute at the originating router.	<b>detail extensive</b>
<b>Layer2-info: encaps</b>	Layer 2 encapsulation (for example, VPLS).	<b>detail extensive</b>
<b>control flags</b>	Control flags: <b>none</b> or <b>Site Down</b> .	<b>detail extensive</b>
<b>mtu</b>	Maximum transmission unit (MTU) of the Layer 2 circuit.	<b>detail extensive</b>

## 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 bgp detail (Layer 3 VPN)

```

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 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

<b>List of Syntax</b>	<a href="#">Syntax on page 897</a> <a href="#">Syntax (EX Series Switches) on page 897</a>
<b>Syntax</b>	show route all <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	show route all
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display information about all routes in all routing tables, including private, or internal, tables.
<b>Options</b>	<b>none</b> —Display information about all routes in all routing tables, including private, or internal, tables.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route all on page 897</a>
<b>Output Fields</b>	In Junos OS Release 9.5 and later, only the output fields for the <b>show route all</b> command display all routing tables, including private, or hidden, routing tables. The output field table of the <b>show route</b> 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 aspath-regex

<b>List of Syntax</b>	<a href="#">Syntax on page 899</a> <a href="#">Syntax (EX Series Switches) on page 899</a>
<b>Syntax</b>	<pre>show route aspath-regex <i>regular-expression</i> &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route aspath-regex <i>regular-expression</i></pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	<p>Display the entries in the routing table that match the specified autonomous system (AS) path regular expression.</p>
<b>Options</b>	<p><b><i>regular-expression</i></b>—Regular expression that matches an entire AS path.</p> <p><b><i>logical-system (all   <i>logical-system-name</i>)</i></b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Additional Information</b>	<p>You can specify a regular expression as:</p> <ul style="list-style-type: none"> <li>• An individual AS number</li> <li>• A period wildcard used in place of an AS number</li> <li>• An AS path regular expression that is enclosed in parentheses</li> </ul> <p>You also can include the operators described in the table of AS path regular expression operators in the <i>Junos Policy Framework Configuration Guide</i>. The following list summarizes these operators:</p> <ul style="list-style-type: none"> <li>• <b><i>{m,n}</i></b>—At least <i>m</i> and at most <i>n</i> repetitions of the AS path term.</li> <li>• <b><i>{m}</i></b>—Exactly <i>m</i> repetitions of the AS path term.</li> <li>• <b><i>{m,}</i></b>—<i>m</i> or more repetitions of the AS path term.</li> <li>• <b><i>*</i></b>—Zero or more repetitions of an AS path term.</li> <li>• <b><i>+</i></b>—One or more repetitions of an AS path term.</li> <li>• <b><i>?</i></b>—Zero or one repetition of an AS path term.</li> <li>• <b><i>aspath_term   aspath_term</i></b>—Match one of the two AS path terms.</li> </ul> <p>When you specify more than one AS number or path term, or when you include an operator in the regular expression, enclose the entire regular expression in quotation marks. For example, to match any path that contains AS number 234, specify the following command:</p> <pre>show route aspath-regex ". * 234 ."</pre>

<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Using AS Path Regular Expressions</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show route aspath-regex (Matching a Specific AS Number) on page 900</a> <a href="#">show route aspath-regex (Matching Any Path with Two AS Numbers) on page 900</a>
<b>Output Fields</b>	For information about output fields, see the output field table for the <a href="#">show route</a> command.

## Sample Output

### show route aspath-regex (Matching a Specific AS Number)

```

user@host> show route aspath-regex 65477
inet.0: 46411 destinations, 46411 routes (46409 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

111.222.1.0/25      *[BGP/170] 00:08:48, localpref 100, from 111.222.2.24
                   AS Path: [65477] ({65488 65535}) IGP
                   to 111.222.18.225 via fpa0.0(111.222.18.233)
111.222.1.128/25   *[IS-IS/15] 09:15:37, metric 37, tag 1
                   to 111.222.18.225 via fpa0.0(111.222.18.233)
                   [BGP/170] 00:08:48, localpref 100, from 111.222.2.24
                   AS Path: [65477] ({65488 65535}) IGP
                   to 111.222.18.225 via fpa0.0(111.222.18.233)
...

```

### show route aspath-regex (Matching Any Path with Two AS Numbers)

```

user@host> show route aspath-regex ?.* 234 3561.*?

inet.0: 46351 destinations, 46351 routes (46349 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

9.20.0.0/17        *[BGP/170] 01:35:00, localpref 100, from 131.103.20.49
                   AS Path: [666] 234 3561 2685 2686 Incomplete
                   to 192.156.169.1 via 192.156.169.14(so-0/0/0)
12.10.231.0/24     *[BGP/170] 01:35:00, localpref 100, from 131.103.20.49
                   AS Path: [666] 234 3561 5696 7369 IGP
                   to 192.156.169.1 via 192.156.169.14(so-0/0/0)
24.64.32.0/19      *[BGP/170] 01:34:59, localpref 100, from 131.103.20.49
                   AS Path: [666] 234 3561 6327 IGP
                   to 192.156.169.1 via 192.156.169.14(so-0/0/0)
...

```

## show route best

<b>List of Syntax</b>	<a href="#">Syntax on page 901</a> <a href="#">Syntax (EX Series Switches) on page 901</a>
<b>Syntax</b>	show route best <i>destination-prefix</i> <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	show route best <i>destination-prefix</i> <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	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.
<b>Options</b>	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b> .  <b>destination-prefix</b> —Address or range of addresses.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route best on page 901</a> <a href="#">show route best detail on page 902</a> <a href="#">show route best extensive on page 903</a> <a href="#">show route best terse on page 903</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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 902](#).

### 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

<b>List of Syntax</b>	<a href="#">Syntax on page 904</a> <a href="#">Syntax (EX Series Switches) on page 904</a>
<b>Syntax</b>	show route brief <destination-prefix> <logical-system (all   logical-system-name)>
<b>Syntax (EX Series Switches)</b>	show route brief <destination-prefix>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display brief information about the active entries in the routing tables.
<b>Options</b>	<b>none</b> —Display all active entries in the routing table.  <b>destination-prefix</b> —(Optional) Display active entries for the specified address or range of addresses.  <b>logical-system (all   logical-system-name)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route brief on page 904</a>
<b>Output Fields</b>	For information about output fields, see the Output Field table of the <a href="#">show route</a> 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 community

---

List of Syntax	<a href="#">Syntax on page 906</a> <a href="#">Syntax (EX Series Switches) on page 906</a>
Syntax	<code>show route community <i>as-number:community-value</i></code> <code>&lt;brief   detail   extensive   terse&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
Syntax (EX Series Switches)	<code>show route community <i>as-number:community-value</i></code> <code>&lt;brief   detail   extensive   terse&gt;</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 entries in each routing table that are members of a Border Gateway Protocol (BGP) community.
Options	<p><b><i>as-number:community-value</i></b>—One or more community identifiers. <b><i>as-number</i></b> is the AS number, and <b><i>community-value</i></b> is the community identifier. When you specify more than one community identifier, enclose the identifiers in double quotation marks. Community identifiers can include wildcards.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
Additional Information	Specifying the community option displays all routes matching the community found within the routing table. The community option does not limit the output to only the routes being advertised to the neighbor after any egress routing policy.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"><li>• <a href="#">show route detail on page 915</a></li></ul>
List of Sample Output	<a href="#">show route community on page 906</a>
Output Fields	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> command.

## Sample Output

### show route community

```
user@host> show route community 234:80
inet.0: 46511 destinations, 46511 routes (46509 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
4.0.0.0/8      *[BGP/170] 03:33:07, localpref 100, from 131.103.20.49
                AS Path: {666} 234 2548 1 IGP
                to 192.156.169.1 via 192.156.169.14(so-0/0/0)
6.0.0.0/8      *[BGP/170] 03:33:07, localpref 100, from 131.103.20.49
                AS Path: {666} 234 2548 568 721 Incomplete
                to 192.156.169.1 via 192.156.169.14(so-0/0/0)
9.2.0.0/16     *[BGP/170] 03:33:06, localpref 100, from 131.103.20.49
                AS Path: {666} 234 2548 1673 1675 1747 IGP
                to 192.156.169.1 via 192.156.169.14(so-0/0/0)
```

## show route community-name

<b>List of Syntax</b>	<a href="#">Syntax on page 908</a> <a href="#">Syntax (EX Series Switches) on page 908</a>
<b>Syntax</b>	<b>show route community-name</b> <i>community-name</i> <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	<b>show route community-name</b> <i>community-name</i> <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display the route entries in each routing table that are members of a Border Gateway Protocol (BGP) community, specified by a community name.
<b>Options</b>	<i>community-name</i> —Name of the community.  <b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route community-name on page 908</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> command.

## Sample Output

### show route community-name

```

user@host> show route community-name red-com
inet.0: 17 destinations, 17 routes (16 active, 0 holddown, 1 hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

instance1.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

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

10.255.245.212/32  *[BGP/170] 00:04:40, localpref 100, from 10.255.245.204
                  AS path: 300 I
                  > to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix
20.20.20.20/32    *[BGP/170] 00:04:40, localpref 100, from 10.255.245.204
                  AS path: I
                  > to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix
100.1.4.0/24     *[BGP/170] 00:04:40, localpref 100, from 10.255.245.204

```

```

AS path: I
> to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix

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: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.245.204:10:10.255.245.212/32
*[BGP/170] 00:06:40, localpref 100, from 10.255.245.204
AS path: 300 I
> to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix
10.255.245.204:10:20.20.20.20/32
*[BGP/170] 00:36:02, localpref 100, from 10.255.245.204
AS path: I
> to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix
10.255.245.204:10:100.1.4.0/24
*[BGP/170] 00:36:02, localpref 100, from 10.255.245.204
AS path: I
> to 100.1.2.2 via ge-1/1/0.0, label-switched-path to_fix

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

instance1.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

```

## show route damping

---

List of Syntax	<a href="#">Syntax on page 910</a> <a href="#">Syntax (EX Series Switch and QFX Series) on page 910</a>
Syntax	<code>show route damping (decayed   history   suppressed)</code> <code>&lt;brief   detail   extensive   terse&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
Syntax (EX Series Switch and QFX Series)	<code>show route damping (decayed   history   suppressed)</code> <code>&lt;brief   detail   extensive   terse&gt;</code>
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 the BGP routes for which updates might have been reduced because of route flap damping.
Options	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.  <b>decayed</b> —Display route damping entries that might no longer be valid, but are not suppressed.  <b>history</b> —Display entries that have already been withdrawn, but have been logged.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.  <b>suppressed</b> —Display entries that have been suppressed and are no longer being installed into the forwarding table or exported by routing protocols.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"><li>• <a href="#">clear bgp damping on page 799</a></li><li>• <a href="#">show policy damping on page 873</a></li></ul>
List of Sample Output	<a href="#">show route damping decayed detail on page 913</a> <a href="#">show route damping history on page 914</a> <a href="#">show route damping history detail on page 914</a>
Output Fields	<a href="#">Table 26 on page 911</a> lists the output fields for the <b>show route damping</b> command. Output fields are listed in the approximate order in which they appear.

Table 26: show route damping Output Fields

Field Name	Field Description	Level of Output
<i>routing-table-name</i>	Name of the routing table—for example, <b>inet.0</b> .	All levels
<b>destinations</b>	Number of destinations for which there are routes in the routing table.	All levels
<b>number routes</b>	Number of routes in the routing table and total number of routes in the following states: <ul style="list-style-type: none"> <li>• <b>active</b></li> <li>• <b>holddown</b> (routes that are in a pending state before being declared inactive)</li> <li>• <b>hidden</b> (the routes are not used because of a routing policy)</li> </ul>	All levels
<b>destination-prefix (entry, announced)</b>	Destination prefix. The <b>entry</b> value is the number of routes for this destination, and the <b>announced</b> value is the number of routes being announced for this destination.	<b>detail extensive</b>
<b>[protocol, preference]</b>	Protocol from which the route was learned and the preference value for the route. <ul style="list-style-type: none"> <li>• <b>+</b>—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table.</li> <li>• <b>-</b>—A hyphen indicates the last active route.</li> <li>• <b>*</b>—An asterisk indicates that the route is both the active and the last active route. An asterisk before a <b>to</b> line indicates the best subpath to the route.</li> </ul> <p>In every routing metric except for the BGP <b>LocalPref</b> attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the <b>LocalPref</b> value in the <b>Preference2</b> field. For example, if the <b>LocalPref</b> value for Route 1 is 100, the <b>Preference2</b> value is -101. If the <b>LocalPref</b> value for Route 2 is 155, the <b>Preference2</b> value is -156. Route 2 is preferred because it has a higher <b>LocalPref</b> value and a lower <b>Preference2</b> value.</p>	All levels
<b>Next-hop reference count</b>	Number of references made to the next hop.	<b>detail extensive</b>
<b>Source</b>	IP address of the route source.	<b>detail extensive</b>
<b>Next hop</b>	Network layer address of the directly reachable neighboring system.	<b>detail extensive</b>
<b>via</b>	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 <b>Selected</b> .	<b>detail extensive</b>
<b>Protocol next hop</b>	Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop.	<b>detail extensive</b>
<b>Indirect next hop</b>	Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops.	<b>detail extensive</b>
<b>State</b>	Flags for this route. For a description of possible values for this field, see the output field table for the <a href="#">show route detail</a> command.	<b>detail extensive</b>

Table 26: show route damping Output Fields (*continued*)

Field Name	Field Description	Level of Output
Local AS	AS number of the local routing device.	detail extensive
Peer AS	AS number of the peer routing device.	detail extensive
Age	How long the route has been known.	detail extensive
Metric	Metric for the route.	detail extensive
Task	Name of the protocol that has added the route.	detail extensive
Announcement bits	List of protocols that announce this route. <i>n-Resolve inet</i> indicates that the route is used for route resolution for next hops found in the routing table. <i>n</i> is an index used by Juniper Networks customer support only.	detail extensive
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"> <li>• I—IGP.</li> <li>• E—EGP.</li> <li>• ?—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• [ ]—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.</li> <li>• { }—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.</li> <li>• ( )—Parentheses enclose a confederation.</li> <li>• ( [ ] )—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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
to	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.	brief none
via	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 <b>Selected</b> .	brief none
Communities	Community path attribute for the route. See the output field table for the <a href="#">show route detail</a> command.	detail extensive
Localpref	Local preference value included in the route.	All levels
Router ID	BGP router ID as advertised by the neighbor in the open message.	detail extensive

Table 26: show route damping Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Merit (last update/now)</b>	Last updated and current figure-of-merit value.	<b>detail extensive</b>
<b>damping-parameters</b>	Name that identifies the damping parameters used, which is defined in the damping statement at the <b>[edit policy-options]</b> hierarchy level.	<b>detail extensive</b>
<b>Last update</b>	Time of most recent change in path attributes.	<b>detail extensive</b>
<b>First update</b>	Time of first change in path attributes, which started the route damping process.	<b>detail extensive</b>
<b>Flaps</b>	Number of times the route has gone up or down or its path attributes have changed.	<b>detail extensive</b>
<b>Suppressed</b>	( <b>suppressed</b> keyword only) This route is currently suppressed. A suppressed route does not appear in the forwarding table and routing protocols do not export it.	All levels
<b>Reusable in</b>	( <b>suppressed</b> keyword only) Time when a suppressed route will again be available.	All levels
<b>Preference will be</b>	( <b>suppressed</b> keyword only) Preference value that will be applied to the route when it is again active.	All levels

## Sample Output

### show route damping decayed detail

```

user@host> show route damping decayed detail
inet.0: 173319 destinations, 1533668 routes (172625 active, 4 holddown, 108083
hidden)
10.0.111.0/24 (7 entries, 1 announced)
  *BGP      Preference: 170/-101
            Next-hop reference count: 151973
            Source: 172.23.2.129
            Next hop: via so-1/2/0.0
            Next hop: via so-5/1/0.0, selected
            Next hop: via so-6/0/0.0
            Protocol next hop: 172.23.2.129
            Indirect next hop: 89a1a00 264185
            State: <Active Ext>
            Local AS: 65000 Peer AS: 65490
            Age: 3:28      Metric2: 0
            Task: BGP_65490.172.23.2.129+179
            Announcement bits (6): 0-KRT 1-RT 4-KRT 5-BGP.0.0.0.0+179

        6-Resolve tree 2 7-Resolve tree 3
        AS path: 65490 65520 65525 65525 65525 65525 I ()
        Communities: 65501:390 65501:2000 65501:3000 65504:701
        Localpref: 100
        Router ID: 172.23.2.129
        Merit (last update/now): 1934/1790
        damping-parameters: damping-high

```

```
Last update:      00:03:28 First update:      00:06:40
Flaps: 2
```

### show route damping history

```
user@host> show route damping history
inet.0: 173320 destinations, 1533529 routes (172624 active, 6 holddown, 108122
hidden)
+ = Active Route, - = Last Active, * = Both

10.108.0.0/15      [BGP ] 2d 22:47:58, localpref 100
                  AS path: 65220 65501 65502 I
                  > to 192.168.60.85 via so-3/1/0.0
```

### show route damping history detail

```
user@host> show route damping history detail
inet.0: 173319 destinations, 1533435 routes (172627 active, 2 holddown, 108105
hidden)
10.108.0.0/15 (3 entries, 1 announced)
    BGP                /-101
        Next-hop reference count: 69058
        Source: 192.168.60.85
        Next hop: 192.168.60.85 via so-3/1/0.0, selected
        State: <Hidden Ext>
        Inactive reason: Unusable path
        Local AS: 65000 Peer AS: 65220
        Age: 2d 22:48:10
        Task: BGP_65220.192.168.60.85+179
        AS path: 65220 65501 65502 I ()
        Communities: 65501:390 65501:2000 65501:3000 65504:3561
        Localpref: 100
        Router ID: 192.168.80.25
        Merit (last update/now): 1000/932
        damping-parameters: set-normal
        Last update:      00:01:05 First update:      00:01:05
        Flaps: 1
```

## show route detail

<b>List of Syntax</b>	<a href="#">Syntax on page 915</a> <a href="#">Syntax (EX Series Switches) on page 915</a>
<b>Syntax</b>	show route detail <destination-prefix> <logical-system (all   logical-system-name)>
<b>Syntax (EX Series Switches)</b>	show route detail <destination-prefix>
<b>Release Information</b>	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 13.2X51-D15 for the QFX Series.
<b>Description</b>	Display detailed information about the active entries in the routing tables.
<b>Options</b>	<b>none</b> —Display all active entries in the routing table on all systems.  <b>destination-prefix</b> —(Optional) Display active entries for the specified address or range of addresses.  <b>logical-system (all   logical-system-name)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route detail on page 924</a> <a href="#">show route detail (with BGP Multipath) on page 930</a> <a href="#">show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 930</a> <a href="#">show route label detail (Multipoint LDP with Multicast-Only Fast Reroute) on page 931</a>
<b>Output Fields</b>	Table 27 on page 915 describes the output fields for the <b>show route detail</b> command. Output fields are listed in the approximate order in which they appear.

Table 27: 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"> <li>• <b>active</b> (routes that are active)</li> <li>• <b>holddown</b> (routes that are in the pending state before being declared inactive)</li> <li>• <b>hidden</b> (routes that are not used because of a routing policy)</li> </ul>

Table 27: 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 <b>entry</b> value is the number of routes for this destination, and the <b>announced</b> 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"> <li>• <b>MPLS-label</b> (for example, 80001).</li> <li>• <b>interface-name</b> (for example, ge-1/0/2).</li> <li>• <b>neighbor-address:control-word-status:encapsulation type:vc-id:source</b> (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96). <ul style="list-style-type: none"> <li>• <b>neighbor-address</b>—Address of the neighbor.</li> <li>• <b>control-word-status</b>—Whether the use of the control word has been negotiated for this virtual circuit: <b>NoCtrlWord</b> or <b>CtrlWord</b>.</li> <li>• <b>encapsulation type</b>—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.</li> <li>• <b>vc-id</b>—Virtual circuit identifier.</li> <li>• <b>source</b>—Source of the advertisement: <b>Local</b> or <b>Remote</b>.</li> </ul> </li> </ul>
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"> <li>• <b>S=0 route</b> 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).</li> <li>• If there is no <b>S=</b> information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed).</li> </ul>
[ <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"> <li>• <b>+—</b>A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table.</li> <li>• <b>- —</b>A hyphen indicates the last active route.</li> <li>• <b>*—</b>An asterisk indicates that the route is both the active and the last active route. An asterisk before a <b>to</b> line indicates the best subpath to the route.</li> </ul> <p>In every routing metric except for the BGP <b>LocalPref</b> attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the <b>LocalPref</b> value in the <b>Preference2</b> field. For example, if the <b>LocalPref</b> value for Route 1 is 100, the <b>Preference2</b> value is -101. If the <b>LocalPref</b> value for Route 2 is 155, the <b>Preference2</b> value is -156. Route 2 is preferred because it has a higher <b>LocalPref</b> value and a lower <b>Preference2</b> 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.
PMSI	Provider multicast service interface (MVPN routing table).
Next-hop type	Type of next hop. For a description of possible values for this field, see <a href="#">Table 28 on page 920</a> .

Table 27: show route detail Output Fields (*continued*)

Field Name	Field Description
<b>Next-hop reference count</b>	Number of references made to the next hop.
<b>Flood nexthop branches exceed maximum message</b>	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.
<b>Source</b>	IP address of the route source.
<b>Next hop</b>	Network layer address of the directly reachable neighboring system.
<b>via</b>	<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 <b>Selected</b>. This field can also contain the following information:</p> <ul style="list-style-type: none"> <li>• <b>Weight</b>—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.</li> <li>• <b>Balance</b>—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.</li> </ul>
<b>Label-switched-path lsp-path-name</b>	Name of the LSP used to reach the next hop.
<b>Label operation</b>	MPLS label and operation occurring at this routing device. The operation can be <b>pop</b> (where a label is removed from the top of the stack), <b>push</b> (where another label is added to the label stack), or <b>swap</b> (where a label is replaced by another label).
<b>Interface</b>	(Local only) Local interface name.
<b>Protocol next hop</b>	Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop.
<b>Indirect next hop</b>	Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops.
<b>State</b>	State of the route (a route can be in more than one state). See <a href="#">Table 29 on page 921</a> .
<b>Local AS</b>	AS number of the local routing device.
<b>Age</b>	How long the route has been known.
<b>AIGP</b>	Accumulated interior gateway protocol (AIGP) BGP attribute.
<b>Metricn</b>	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 27: show route detail Output Fields (*continued*)

Field Name	Field Description
<b>MED-plus-IGP</b>	Metric value for BGP path selection to which the IGP cost to the next-hop destination has been added.
<b>TTL-Action</b>	<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 <a href="#">show route table</a>.</p>
<b>Task</b>	Name of the protocol that has added the route.
<b>Announcement bits</b>	List of protocols that announce this route. <b>n-Resolve inet</b> indicates that the route is used for route resolution for next hops found in the routing table. <b>n</b> is an index used by Juniper Networks customer support only.
<b>AS path</b>	<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"> <li>• <b>I</b>—IGP.</li> <li>• <b>E</b>—EGP.</li> <li>• <b>Recorded</b>—The AS path is recorded by the sample process (sampled).</li> <li>• <b>?</b>—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• <b>[ ]</b>—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.</li> <li>• <b>[ ]</b>—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.</li> <li>• <b>{ }</b>—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.</li> <li>• <b>( )</b>—Parentheses enclose a confederation.</li> <li>• <b>( [ ] )</b>—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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>
<b>validation-state</b>	<p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> <li>• <b>Invalid</b>—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.</li> <li>• <b>Unknown</b>—Indicates that the prefix is not among the prefixes or prefix ranges in the database.</li> <li>• <b>Unverified</b>—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers.</li> <li>• <b>Valid</b>—Indicates that the prefix and autonomous system pair are found in the database.</li> </ul>
<b>FECs bound to route</b>	Point-to-multipoint root address, multicast source address, and multicast group address when multipoint LDP (M-LDP) inband signaling is configured.

Table 27: show route detail Output Fields (*continued*)

Field Name	Field Description
Primary Upstream	When multipoint LDP with multicast-only fast reroute (MoFRR) is configured, the primary upstream path. MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path.
RPF Nexthops	When multipoint LDP with MoFRR is configured, the reverse-path forwarding (RPF) next-hop information. Data packets are received from both the primary path and the secondary paths. The redundant packets are discarded at topology merge points due to the RPF checks.
Label	Multiple MPLS labels are used to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check. Only the primary label is forwarded while all others are dropped. Multiple interfaces can receive packets using the same label.
weight	Value used to distinguish MoFRR primary and backup routes. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible.
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 <a href="#">Table 30 on page 923</a> for all possible values for this field.
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).
control flags	Control flags: <b>none</b> or <b>Site Down</b> .
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 28 on page 920 describes all possible values for the Next-hop Types output field.

**Table 28: Next-hop Types Output Field Values**

Next-Hop Type	Description
<b>Broadcast (bcast)</b>	Broadcast next hop.
<b>Deny</b>	Deny next hop.
<b>Discard</b>	Discard next hop.
<b>Flood</b>	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.
<b>Hold</b>	Next hop is waiting to be resolved into a unicast or multicast type.
<b>Indexed (idxd)</b>	Indexed next hop.
<b>Indirect (indr)</b>	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.
<b>Interface</b>	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.
<b>Local (locl)</b>	Local address on an interface. This next-hop type causes packets with this destination address to be received locally.
<b>Multicast (mcst)</b>	Wire multicast next hop (limited to the LAN).
<b>Multicast discard (mdsc)</b>	Multicast discard.
<b>Multicast group (mgrp)</b>	Multicast group member.
<b>Receive (recv)</b>	Receive.
<b>Reject (rjct)</b>	Discard. An ICMP unreachable message was sent.
<b>Resolve (rslv)</b>	Resolving next hop.
<b>Routed multicast (mcrt)</b>	Regular multicast next hop.

Table 28: Next-hop Types Output Field Values (*continued*)

Next-Hop Type	Description
<b>Router</b>	<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"> <li>• Must not be a direct or local subnet for the routing device.</li> <li>• Must have a next hop that is directly connected to the routing device.</li> </ul>
<b>Table</b>	Routing table next hop.
<b>Unicast (ucst)</b>	Unicast.
<b>Unilist (ulst)</b>	List of unicast next hops. A packet sent to this next hop goes to any next hop in the list.

Table 29 on page 921 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 29: State Output Field Values

Value	Description
<b>Accounting</b>	Route needs accounting.
<b>Active</b>	Route is active.
<b>Always Compare MED</b>	Path with a lower multiple exit discriminator (MED) is available.
<b>AS path</b>	Shorter AS path is available.
<b>Cisco Non-deterministic MED selection</b>	Cisco nondeterministic MED is enabled, and a path with a lower MED is available.
<b>Clone</b>	Route is a clone.
<b>Cluster list length</b>	Length of cluster list sent by the route reflector.
<b>Delete</b>	Route has been deleted.
<b>Ex</b>	Exterior route.
<b>Ext</b>	BGP route received from an external BGP neighbor.

Table 29: State Output Field Values (*continued*)

Value	Description
<b>FlashAll</b>	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.
<b>Hidden</b>	Route not used because of routing policy.
<b>IfCheck</b>	Route needs forwarding RPF check.
<b>IGP metric</b>	Path through next hop with lower IGP metric is available.
<b>Inactive reason</b>	Flags for this route, which was not selected as best for a particular destination.
<b>Initial</b>	Route being added.
<b>Int</b>	Interior route.
<b>Int Ext</b>	BGP route received from an internal BGP peer or a BGP confederation peer.
<b>Interior &gt; Exterior &gt; Exterior via Interior</b>	Direct, static, IGP, or EBGP path is available.
<b>Local Preference</b>	Path with a higher local preference value is available.
<b>Martian</b>	Route is a martian (ignored because it is obviously invalid).
<b>MartianOK</b>	Route exempt from martian filtering.
<b>Next hop address</b>	Path with lower metric next hop is available.
<b>No difference</b>	Path from neighbor with lower IP address is available.
<b>NoReadvrt</b>	Route not to be advertised.
<b>NotBest</b>	Route not chosen because it does not have the lowest MED.
<b>Not Best in its group</b>	Incoming BGP AS is not the best of a group (only one AS can be the best).
<b>NotInstall</b>	Route not to be installed in the forwarding table.
<b>Number of gateways</b>	Path with a greater number of next hops is available.
<b>Origin</b>	Path with a lower origin code is available.
<b>Pending</b>	Route pending because of a hold-down configured on another route.

Table 29: State Output Field Values (*continued*)

Value	Description
<b>Release</b>	Route scheduled for release.
<b>RIB preference</b>	Route from a higher-numbered routing table is available.
<b>Route Distinguisher</b>	64-bit prefix added to IP subnets to make them unique.
<b>Route Metric or MED comparison</b>	Route with a lower metric or MED is available.
<b>Route Preference</b>	Route with lower preference value is available
<b>Router ID</b>	Path through a neighbor with lower ID is available.
<b>Secondary</b>	Route not a primary route.
<b>Unusable path</b>	Path is not usable because of one of the following conditions: <ul style="list-style-type: none"> <li>• The route is damped.</li> <li>• The route is rejected by an import policy.</li> <li>• The route is unresolved.</li> </ul>
<b>Update source</b>	Last tiebreaker is the lowest IP address value.

Table 30 on page 923 describes the possible values for the Communities output field.

Table 30: 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.
<b>bandwidth: local AS number:link-bandwidth-number</b>	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.
<b>domain-id</b>	Unique configurable number that identifies the OSPF domain.
<b>domain-id-vendor</b>	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.

Table 30: Communities Output Field Values (*continued*)

Value	Description
<b>origin</b>	(Used with VPNs) Identifies where the route came from.
<b>ospf-route-type</b>	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.
<b>route-type-vendor</b>	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute <b>0x8000</b> . The format is <b>area-number:ospf-route-type:options</b> .
<b>rte-type</b>	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute <b>0x0306</b> . The format is <b>area-number:ospf-route-type:options</b> .
<b>target</b>	Defines which VPN the route participates in; <b>target</b> has the format <b>32-bit IP address:16-bit number</b> . For example, 10.19.0.0:100.
<b>unknown IANA</b>	Incoming IANA codes with a value between <b>0x1</b> and <b>0x7fff</b> . This code of the BGP extended community attribute is accepted, but it is not recognized.
<b>unknown OSPF vendor community</b>	Incoming IANA codes with a value above <b>0x8000</b> . 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)

299840 (1 entry, 1 announced)

```

```

TSI:
KRT in-kerne 299840 /52 -> {indirect(1048575)}
    *RSVP Preference: 7/2
        Next hop type: Flood
        Address: 0x9174a30
        Next-hop reference count: 4
        Next hop type: Router, Next hop index: 798
        Address: 0x9174c28
        Next-hop reference count: 2
        Next hop: 8.0.0.2 via lt-1/2/0.9 weight 0x1
        Label-switched-path R2-to-R4-2p2mp
        Label operation: Pop
        Next hop type: Router, Next hop index: 1048574
        Address: 0x92544f0
        Next-hop reference count: 2
        Next hop: 7.0.0.2 via lt-1/2/0.7 weight 0x1
        Label-switched-path R2-to-R200-p2mp
        Label operation: Pop
        Next hop: 6.0.0.2 via lt-1/2/0.5 weight 0x8001
        Label operation: Pop
        State: <Active Int>
        Age: 1:29 Metric: 1
        Task: RSVP
        Announcement bits (1): 0-KRT
        AS path: I...

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

...

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: l2 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
        Validation State: unverified
        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
        Validation State: unverified
        Task: BGP_2.10.1.1.6+58198
        AS path: 2 I
        Accepted MultipathContrib
        Localpref: 100
        Router ID: 1.1.1.3

```

### show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```

user@host> show route label 299872 detail
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
299872 (1 entry, 1 announced)
  *LDP   Preference: 9
        Next hop type: Flood
        Next-hop reference count: 3
        Address: 0x9097d90
        Next hop: via vt-0/1/0.1
        Next-hop index: 661
        Label operation: Pop
        Address: 0x9172130
        Next hop: via so-0/0/3.0
        Next-hop index: 654
        Label operation: Swap 299872
        State: **Active Int>
        Local AS: 1001

```

```

Age: 8:20      Metric: 1
Task: LDP
Announcement bits (1): 0-KRT
AS path: I
FECs bound to route: P2MP root-addr 10.255.72.166, grp 232.1.1.1,
src 192.168.142.2

```

### show route label detail (Multipoint LDP with Multicast-Only Fast Reroute)

```
user@host> show route label 301568 detail
```

```

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
  *LDP    Preference: 9
    Next hop type: Flood
    Address: 0x2735208
    Next-hop reference count: 3
    Next hop type: Router, Next hop index: 1397
    Address: 0x2735d2c
    Next-hop reference count: 3
    Next hop: 1.3.8.2 via ge-1/2/22.0
    Label operation: Pop
    Load balance label: None;
    Next hop type: Router, Next hop index: 1395
    Address: 0x2736290
    Next-hop reference count: 3
    Next hop: 1.3.4.2 via ge-1/2/18.0
    Label operation: Pop
    Load balance label: None;
    State: <Active Int AckRequest MulticastRPF>
    Local AS: 10
    Age: 54:05      Metric: 1
    Validation State: unverified
    Task: LDP
    Announcement bits (1): 0-KRT
    AS path: I
    FECs bound to route: P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src:
192.168.219.11
      Primary Upstream : 1.1.1.3:0--1.1.1.2:0
        RPF Nexthops :
          ge-1/2/15.0, 1.2.94.1, Label: 301568, weight: 0x1
          ge-1/2/14.0, 1.2.3.1, Label: 301568, weight: 0x1
      Backup Upstream : 1.1.1.3:0--1.1.1.6:0
        RPF Nexthops :
          ge-1/2/20.0, 1.2.96.1, Label: 301584, weight: 0xffffe
          ge-1/2/19.0, 1.3.6.1, Label: 301584, weight: 0xffffe

```

## show route exact

---

List of Syntax	<a href="#">Syntax on page 932</a> <a href="#">Syntax (EX Series Switches) on page 932</a>
Syntax	<code>show route exact <i>destination-prefix</i></code> <code>&lt;brief   detail   extensive   terse&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
Syntax (EX Series Switches)	<code>show route exact <i>destination-prefix</i></code> <code>&lt;brief   detail   extensive   terse&gt;</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	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b> .  <b><i>destination-prefix</i></b> —Address or range of addresses.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
List of Sample Output	<a href="#">show route exact on page 932</a> <a href="#">show route exact detail on page 932</a> <a href="#">show route exact extensive on page 933</a> <a href="#">show route exact terse on page 933</a>
Output Fields	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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

<b>List of Syntax</b>	<a href="#">Syntax on page 934</a> <a href="#">Syntax (EX Series Switches) on page 934</a>
<b>Syntax</b>	<pre>show route export &lt;brief   detail&gt; &lt;instance &lt;instance-name&gt;   routing-table-name&gt; &lt;logical-system (all   logical-system-name)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route export &lt;brief   detail&gt; &lt;instance &lt;instance-name&gt;   routing-table-name&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display policy-based route export information. Policy-based export simplifies the process of exchanging route information between routing instances.
<b>Options</b>	<p><b>none</b>—(Same as <b>brief</b>.) Display standard information about policy-based export for all instances and routing tables on all systems.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output.</p> <p><b>instance &lt;instance-name&gt;</b>—(Optional) Display a particular routing instance for which policy-based export is currently enabled.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>routing-table-name</b>—(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 <b>show route export inet</b> command).</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route export on page 935</a> <a href="#">show route export detail on page 935</a> <a href="#">show route export instance detail on page 935</a>
<b>Output Fields</b>	<a href="#">Table 31 on page 934</a> lists the output fields for the <b>show route export</b> command. Output fields are listed in the approximate order in which they appear.

**Table 31: show route export Output Fields**

Field Name	Field Description	Level of Output
Table or <i>table-name</i>	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.	<b>brief</b> none

Table 31: show route export Output Fields (*continued*)

Field Name	Field Description	Level of Output
Export	Whether the table is currently exporting routes to other tables: <b>Y</b> or <b>N</b> (Yes or No).	<b>brief</b> none
Import	Tables currently importing routes from the originator table. (Not displayed for tables that are not exporting any routes.)	<b>detail</b>
Flags	( <b>instance</b> keyword only) Flags for this feature on this instance: <ul style="list-style-type: none"> <li><b>config auto-policy</b>—The policy was deduced from the configured IGP export policies.</li> <li><b>cleanup</b>—Configuration information for this instance is no longer valid.</li> <li><b>config</b>—The instance was explicitly configured.</li> </ul>	<b>detail</b>
Options	( <b>instance</b> keyword only) Configured option displays the type of routing tables the feature handles: <ul style="list-style-type: none"> <li><b>unicast</b>—Indicates <i>instance.inet.0</i>.</li> <li><b>multicast</b>—Indicates <i>instance.inet.2</i>.</li> <li><b>unicast multicast</b>—Indicates <i>instance.inet.0</i> and <i>instance.inet.2</i>.</li> </ul>	<b>detail</b>
Import policy	( <b>instance</b> keyword only) Policy that <b>route export</b> uses to construct the import-export matrix. Not displayed if the instance type is <b>vrf</b> .	<b>detail</b>
Instance	( <b>instance</b> keyword only) Name of the routing instance.	<b>detail</b>
Type	( <b>instance</b> keyword only) Type of routing instance: <b>forwarding</b> , <b>non-forwarding</b> , or <b>vrf</b> .	<b>detail</b>

## 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  
Instance: red

Type: non-forwarding  
Type: non-forwarding

## show route extensive

<b>List of Syntax</b>	<a href="#">Syntax on page 937</a> <a href="#">Syntax (EX Series Switches) on page 937</a>
<b>Syntax</b>	show route extensive <destination-prefix> <logical-system (all   logical-system-name)>
<b>Syntax (EX Series Switches)</b>	show route extensive <destination-prefix>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display extensive information about the active entries in the routing tables.
<b>Options</b>	<b>none</b> —Display all active entries in the routing table.  <b>destination-prefix</b> —(Optional) Display active entries for the specified address or range of addresses.  <b>logical-system (all   logical-system-name)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route extensive on page 944</a> <a href="#">show route extensive (Access Route) on page 950</a> <a href="#">show route extensive (BGP PIC Edge) on page 951</a> <a href="#">show route extensive (FRR and LFA) on page 951</a> <a href="#">show route extensive (Route Reflector) on page 952</a> <a href="#">show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 952</a> <a href="#">show route label detail (Multipoint LDP with Multicast-Only Fast Reroute) on page 953</a>
<b>Output Fields</b>	<a href="#">Table 32 on page 937</a> describes the output fields for the <b>show route extensive</b> command. Output fields are listed in the approximate order in which they appear.

**Table 32: 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.

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
<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"> <li>• <b>active</b> (routes that are active).</li> <li>• <b>holddown</b> (routes that are in the pending state before being declared inactive).</li> <li>• <b>hidden</b> (routes that are not used because of a routing policy).</li> </ul>
<i>route-destination</i> (entry, announced)	<p>Route destination (for example: 10.0.0.1/24). The <b>entry</b> value is the number of route for this destination, and the <b>announced</b> 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"> <li>• <b>MPLS-label</b> (for example, 80001).</li> <li>• <b>interface-name</b> (for example, ge-1/0/2).</li> <li>• <b>neighbor-address:control-word-status:encapsulation type:vc-id:source</b> (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96). <ul style="list-style-type: none"> <li>• <b>neighbor-address</b>—Address of the neighbor.</li> <li>• <b>control-word-status</b>—Whether the use of the control word has been negotiated for this virtual circuit: <b>NoCtrlWord</b> or <b>CtrlWord</b>.</li> <li>• <b>encapsulation type</b>—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.</li> <li>• <b>vc-id</b>—Virtual circuit identifier.</li> <li>• <b>source</b>—Source of the advertisement: <b>Local</b> or <b>Remote</b>.</li> </ul> </li> </ul>
<b>TSI</b>	Protocol header information.
<b>label stacking</b>	<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"> <li>• <b>S=0 route</b> 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).</li> <li>• If there is no <b>S=</b> information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed).</li> </ul>
<b>[protocol, preference]</b>	<p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> <li>• <b>+</b>—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table.</li> <li>• <b>-</b>—A hyphen indicates the last active route.</li> <li>• <b>*</b>—An asterisk indicates that the route is both the active and the last active route. An asterisk before a <b>to</b> line indicates the best subpath to the route.</li> </ul> <p>In every routing metric except for the BGP <b>LocalPref</b> attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the <b>LocalPref</b> value in the <b>Preference2</b> field. For example, if the <b>LocalPref</b> value for Route 1 is 100, the <b>Preference2</b> value is -101. If the <b>LocalPref</b> value for Route 2 is 155, the <b>Preference2</b> value is -156. Route 2 is preferred because it has a higher <b>LocalPref</b> value and a lower <b>Preference2</b> value.</p>

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
<b>Level</b>	(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.
<b>Route Distinguisher</b>	IP subnet augmented with a 64-bit prefix.
<b>PMSI</b>	Provider multicast service interface (MVPN routing table).
<b>Next-hop type</b>	Type of next hop. For a description of possible values for this field, see the Output Field table in the <a href="#">show route detail</a> command.
<b>Next-hop reference count</b>	Number of references made to the next hop.
<b>Flood nexthop branches exceed maximum message</b>	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.
<b>Source</b>	IP address of the route source.
<b>Next hop</b>	Network layer address of the directly reachable neighboring system.
<b>via</b>	<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 <b>Selected</b>. This field can also contain the following information:</p> <ul style="list-style-type: none"> <li>• <b>Weight</b>—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.</li> <li>• <b>Balance</b>—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.</li> </ul>
<b>Label-switched-path lsp-path-name</b>	Name of the LSP used to reach the next hop.
<b>Label operation</b>	MPLS label and operation occurring at this routing device. The operation can be <b>pop</b> (where a label is removed from the top of the stack), <b>push</b> (where another label is added to the label stack), or <b>swap</b> (where a label is replaced by another label).
<b>Offset</b>	Whether the metric has been increased or decreased by an offset value.
<b>Interface</b>	(Local only) Local interface name.
<b>Protocol next hop</b>	Network layer address of the remote routing device that advertised the prefix. This address is used to recursively derive a forwarding next hop.

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
<b><i>label-operation</i></b>	MPLS label and operation occurring at this routing device. The operation can be <b>pop</b> (where a label is removed from the top of the stack), <b>push</b> (where another label is added to the label stack), or <b>swap</b> (where a label is replaced by another label).
<b>Indirect next hops</b>	<p>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.</p> <p>When BGP PIC Edge is enabled, the output lines that contain <b>Indirect next hop: weight</b> follow next hops that the software can use to repair paths where a link failure occurs. The next-hop weight has one of the following values:</p> <ul style="list-style-type: none"> <li>• 0x1 indicates active next hops.</li> <li>• 0x4000 indicates passive next hops.</li> </ul>
<b>State</b>	State of the route (a route can be in more than one state). See the Output Field table in the <a href="#">show route detail</a> command.
<b>Session ID</b>	The BFD session ID number that represents the protection using MPLS fast reroute (FRR) and loop-free alternate (LFA).
<b>Weight</b>	<p>Weight for the backup path. If the weight of an indirect next hop is larger than zero, the weight value is shown.</p> <p>For sample output, see <a href="#">show route table</a>.</p>

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
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"> <li>• <b>Active preferred</b>—Currently active route was selected over this route.</li> <li>• <b>Always compare MED</b>—Path with a lower multiple exit discriminator (MED) is available.</li> <li>• <b>AS path</b>—Shorter AS path is available.</li> <li>• <b>Cisco Non-deterministic MED selection</b>—Cisco nondeterministic MED is enabled and a path with a lower MED is available.</li> <li>• <b>Cluster list length</b>—Path with a shorter cluster list length is available.</li> <li>• <b>Forwarding use only</b>—Path is only available for forwarding purposes.</li> <li>• <b>IGP metric</b>—Path through the next hop with a lower IGP metric is available.</li> <li>• <b>IGP metric type</b>—Path with a lower OSPF link-state advertisement type is available.</li> <li>• <b>Interior &gt; Exterior &gt; Exterior via Interior</b>—Direct, static, IGP, or EBGP path is available.</li> <li>• <b>Local preference</b>—Path with a higher local preference value is available.</li> <li>• <b>Next hop address</b>—Path with a lower metric next hop is available.</li> <li>• <b>No difference</b>—Path from a neighbor with a lower IP address is available.</li> <li>• <b>Not Best in its group</b>—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).</li> <li>• <b>Number of gateways</b>—Path with a higher number of next hops is available.</li> <li>• <b>Origin</b>—Path with a lower origin code is available.</li> <li>• <b>OSPF version</b>—Path does not support the indicated OSPF version.</li> <li>• <b>RIB preference</b>—Route from a higher-numbered routing table is available.</li> <li>• <b>Route distinguisher</b>—64-bit prefix added to IP subnets to make them unique.</li> <li>• <b>Route metric or MED comparison</b>—Route with a lower metric or MED is available.</li> <li>• <b>Route preference</b>—Route with a lower preference value is available.</li> <li>• <b>Router ID</b>—Path through a neighbor with a lower ID is available.</li> <li>• <b>Unusable path</b>—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.</li> <li>• <b>Update source</b>—Last tiebreaker is the lowest IP address value.</li> </ul>
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.
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 <a href="#">show route table</a>.</p>

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
<b>Task</b>	Name of the protocol that has added the route.
<b>Announcement bits</b>	List of protocols that announce this route. <b>n-Resolve inet</b> indicates that the route is used for route resolution for next hops found in the routing table. <b>n</b> is an index used by Juniper Networks customer support only.
<b>AS path</b>	<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"> <li>• <b>I</b>—IGP.</li> <li>• <b>E</b>—EGP.</li> <li>• <b>Recorded</b>—The AS path is recorded by the sample process (sampled).</li> <li>• <b>?</b>—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• <b>[ ]</b>—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.</li> <li>• <b>{ }</b>—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.</li> <li>• <b>( )</b>—Parentheses enclose a confederation.</li> <li>• <b>( [ ] )</b>—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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>
<b>validation-state</b>	<p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> <li>• <b>Invalid</b>—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.</li> <li>• <b>Unknown</b>—Indicates that the prefix is not among the prefixes or prefix ranges in the database.</li> <li>• <b>Unverified</b>—Indicates that origin validation is not enabled for the BGP peers.</li> <li>• <b>Valid</b>—Indicates that the prefix and autonomous system pair are found in the database.</li> </ul>
<b>FECs bound to route</b>	Point-to-multipoint root address, multicast source address, and multicast group address when multipoint LDP (M-LDP) inband signaling is configured.
<b>AS path: I &lt;Originator&gt;</b>	(For route reflected output only) Originator ID attribute set by the route reflector.
<b>Primary Upstream</b>	When multipoint LDP with multicast-only fast reroute (MoFRR) is configured, the primary upstream path. MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path.
<b>RPF Nexthops</b>	When multipoint LDP with MoFRR is configured, the reverse-path forwarding (RPF) next-hop information. Data packets are received from both the primary path and the secondary paths. The redundant packets are discarded at topology merge points due to the RPF checks.

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
Label	Multiple MPLS labels are used to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check. Only the primary label is forwarded while all others are dropped. Multiple interfaces can receive packets using the same label.
weight	Value used to distinguish MoFRR primary and backup routes. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible.
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 <a href="#">show route detail</a> command for all possible values for this field.
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).
control flags	Control flags: <b>none</b> 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.

Table 32: show route extensive Output Fields (*continued*)

Field Name	Field Description
<b>Forwarding nexthops</b>	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-kerne1 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-kerne1 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-kerne1 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)
299840 (1 entry, 1 announced)
TSI:
KRT in-kernel 299840 /52 -> {indirect(1048575)}
  *RSVP   Preference: 7/2
          Next hop type: Flood
          Address: 0x9174a30
          Next-hop reference count: 4
          Next hop type: Router, Next hop index: 798
          Address: 0x9174c28
          Next-hop reference count: 2
          Next hop: 8.0.0.2 via lt-1/2/0.9 weight 0x1
          Label-switched-path R2-to-R4-2p2mp

```

```

Label operation: Pop
Next hop type: Router, Next hop index: 1048574
Address: 0x92544f0
Next-hop reference count: 2
Next hop: 7.0.0.2 via lt-1/2/0.7 weight 0x1
Label-switched-path R2-to-R200-p2mp
Label operation: Pop
Next hop: 6.0.0.2 via lt-1/2/0.5 weight 0x8001
Label operation: Pop
State: <Active Int>
Age: 1:29      Metric: 1
Task: RSVP
Announcement bits (1): 0-KRT
AS path: I...

```

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-kernel 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-kernel 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.0.0.0.0+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.0.0.0.0+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 (BGP PIC Edge)

```

user@host> show route 1.1.1.6 extensive
ed.inet.0: 6 destinations, 9 routes (6 active, 0 holddown, 0 hidden)
  1.1.1.6/32 (3 entries, 2 announced)
    State: <CalcForwarding>
TSI:
KRT in-kerne1 1.1.1.6/32 -> {indirect(1048574), indirect(1048577)}
Page 0 idx 0 Type 1 val 9219e30
  Nexthop: Self
  AS path: [2] 3 I
  Communities: target:2:1
  Path 1.1.1.6 from 1.1.1.4 Vector len 4. Val: 0
..
    #Multipath Preference: 255
      Next hop type: Indirect
      Address: 0x93f4010
      Next-hop reference count: 2
..
      Protocol next hop: 1.1.1.4
      Push 299824
      Indirect next hop: 944c000 1048574 INH Session ID: 0x3
      Indirect next hop: weight 0x1
      Protocol next hop: 1.1.1.5
      Push 299824
      Indirect next hop: 944c1d8 1048577 INH Session ID: 0x4
      Indirect next hop: weight 0x4000
      State: <ForwardingOnly Int Ext>
      Inactive reason: Forwarding use only
      Age: 25      Metric2: 15
      Validation State: unverified
      Task: RT
      Announcement bits (1): 0-KRT
      AS path: 3 I
      Communities: target:2:1

```

## 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-kerne1 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 (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

```

#### show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```

user@host> show route label 299872 detail
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
299872 (1 entry, 1 announced)
*LDP Preference: 9
Next hop type: Flood
Next-hop reference count: 3
Address: 0x9097d90
Next hop: via vt-0/1/0.1
Next-hop index: 661
Label operation: Pop
Address: 0x9172130
Next hop: via so-0/0/3.0
Next-hop index: 654
Label operation: Swap 299872
State: **Active Int>

```

```

Local AS: 1001
Age: 8:20      Metric: 1
Task: LDP
Announcement bits (1): 0-KRT
AS path: I
FECs bound to route: P2MP root-addr 10.255.72.166, grp 232.1.1.1,
src 192.168.142.2

```

### show route label detail (Multipoint LDP with Multicast-Only Fast Reroute)

```
user@host> show route label 301568 detail
```

```

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
  *LDP    Preference: 9
    Next hop type: Flood
    Address: 0x2735208
    Next-hop reference count: 3
    Next hop type: Router, Next hop index: 1397
    Address: 0x2735d2c
    Next-hop reference count: 3
    Next hop: 1.3.8.2 via ge-1/2/22.0
    Label operation: Pop
    Load balance label: None;
    Next hop type: Router, Next hop index: 1395
    Address: 0x2736290
    Next-hop reference count: 3
    Next hop: 1.3.4.2 via ge-1/2/18.0
    Label operation: Pop
    Load balance label: None;
    State: <Active Int AckRequest MulticastRPF>
    Local AS: 10
    Age: 54:05      Metric: 1
    Validation State: unverified
    Task: LDP
    Announcement bits (1): 0-KRT
    AS path: I
    FECs bound to route: P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src:
192.168.219.11
      Primary Upstream : 1.1.1.3:0--1.1.1.2:0
        RPF Nexthops :
          ge-1/2/15.0, 1.2.94.1, Label: 301568, weight: 0x1
          ge-1/2/14.0, 1.2.3.1, Label: 301568, weight: 0x1
      Backup Upstream : 1.1.1.3:0--1.1.1.6:0
        RPF Nexthops :
          ge-1/2/20.0, 1.2.96.1, Label: 301584, weight: 0xffffe
          ge-1/2/19.0, 1.3.6.1, Label: 301584, weight: 0xffffe

```

## show route flow validation

<b>List of Syntax</b>	<a href="#">Syntax on page 954</a> <a href="#">Syntax (EX Series Switches) on page 954</a>
<b>Syntax</b>	<pre>show route flow validation &lt;brief   detail&gt; &lt;ip-prefix&gt; &lt;table table-name&gt; &lt;logical-system (all   logical-system-name)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route flow validation &lt;brief   detail&gt; &lt;ip-prefix&gt; &lt;table table-name&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display flow route information.
<b>Options</b>	<p><b>none</b>—Display flow route information.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.</p> <p><b>ip-prefix</b>—(Optional) IP address for the flow route.</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>table table-name</b>—(Optional) Display flow route information for all routing tables whose name begins with this string (for example, inet.0 and inet6.0 are both displayed when you run the <b>show route flow validation inet</b> command).</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route flow validation on page 955</a>
<b>Output Fields</b>	<p><a href="#">Table 33 on page 954</a> lists the output fields for the <b>show route flow validation</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 33: show route flow validation 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>prefix</i>	Route address.	All levels
Active unicast route	Active route in the routing table.	All levels

Table 33: show route flow validation Output Fields (*continued*)

Field Name	Field Description	Level of Output
Dependent flow destinations	Number of flows for which there are routes in the routing table.	All levels
Origin	Source of the route flow.	All levels
Neighbor AS	Autonomous system identifier of the neighbor.	All levels
Flow destination	Number of entries and number of destinations that match the route flow.	All levels
Unicast best match	Destination that is the best match for the route flow.	All levels
Flags	Information about the route flow.	All levels

## Sample Output

### show route flow validation

```

user@host> show route flow validation
inet.0:
10.0.5.0/24Active unicast route
Dependent flow destinations: 1
Origin: 192.168.224.218, Neighbor AS: 65001
Flow destination (3 entries, 1 match origin)
Unicast best match: 10.0.5.0/24
Flags: SubtreeApex Consistent

```

## show route forwarding-table

<b>List of Syntax</b>	<a href="#">Syntax on page 956</a> <a href="#">Syntax (MX Series Routers) on page 956</a> <a href="#">Syntax (TX Matrix and TX Matrix Plus Routers) on page 956</a>
<b>Syntax</b>	<pre>show route forwarding-table &lt;detail   extensive   summary&gt; &lt;all&gt; &lt;ccc interface-name&gt; &lt;destination destination-prefix&gt; &lt;family family   matching matching&gt; &lt;interface-name interface-name&gt; &lt;label name&gt; &lt;matching matching&gt; &lt;multicast&gt; &lt;table (default   logical-system-name/routing-instance-name   routing-instance-name)&gt; &lt;vlan (all   vlan-name)&gt; &lt;vpn vpn&gt;</pre>
<b>Syntax (MX Series Routers)</b>	<pre>show route forwarding-table &lt;detail   extensive   summary&gt; &lt;all&gt; &lt;bridge-domain (all   domain-name)&gt; &lt;ccc interface-name&gt; &lt;destination destination-prefix&gt; &lt;family family   matching matching&gt; &lt;interface-name interface-name&gt; &lt;label name&gt; &lt;learning-vlan-id learning-vlan-id&gt; &lt;matching matching&gt; &lt;multicast&gt; &lt;table (default   logical-system-name/routing-instance-name   routing-instance-name)&gt; &lt;vlan (all   vlan-name)&gt; &lt;vpn vpn&gt;</pre>
<b>Syntax (TX Matrix and TX Matrix Plus Routers)</b>	<pre>show route forwarding-table &lt;detail   extensive   summary&gt; &lt;all&gt; &lt;ccc interface-name&gt; &lt;destination destination-prefix&gt; &lt;family family   matching matching&gt; &lt;interface-name interface-name&gt; &lt;matching matching&gt; &lt;label name&gt; &lt;lcc number&gt; &lt;multicast&gt; &lt;table routing-instance-name&gt; &lt;vpn vpn&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Option <b>bridge-domain</b> introduced in Junos OS Release 7.5</p> <p>Option <b>learning-vlan-id</b> introduced in Junos OS Release 8.4</p>

Options **all** and **vlan** introduced in Junos OS Release 9.6.  
 Command introduced in Junos OS Release 11.3 for the QFX Series.

**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**—(TX Matrix and TX matrix Plus routers only) (Optional) On a routing matrix composed of a TX Matrix router and T640 routers, 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 or T4000 routers, display information for the specified router (line-card chassis) connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

**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 961](#)  
[show route forwarding-table detail on page 962](#)  
[show route forwarding-table destination extensive \(Weights and Balances\) on page 962](#)  
[show route forwarding-table extensive on page 963](#)  
[show route forwarding-table extensive \(RPF\) on page 964](#)  
[show route forwarding-table family mpls on page 965](#)  
[show route forwarding-table family vpls on page 965](#)  
[show route forwarding-table vpls \(Broadcast, unknown unicast, and multicast \(BUM\) hashing is enabled\) on page 965](#)  
[show route forwarding-table vpls \(Broadcast, unknown unicast, and multicast \(BUM\) hashing is enabled with MAC Statistics\) on page 966](#)  
[show route forwarding-table family vpls extensive on page 966](#)  
[show route forwarding-table table default on page 967](#)  
[show route forwarding-table table logical-system-name/routing-instance-name on page 968](#)

[show route forwarding-table vpn on page 969](#)

**Output Fields** [Table 34 on page 959](#) 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 34: 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 <b>table logical-system-name/routing-instance-name</b> 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.	<b>detail extensive</b>
Route Type (Type)	How the route was placed into the forwarding table. When the <b>detail</b> keyword is used, the route type might be abbreviated (as shown in parentheses): <ul style="list-style-type: none"> <li><b>cloned (clon)</b>—(TCP or multicast only) Cloned route.</li> <li><b>destination (dest)</b>—Remote addresses directly reachable through an interface.</li> <li><b>destination down (iddn)</b>—Destination route for which the interface is unreachable.</li> <li><b>interface cloned (ifcl)</b>—Cloned route for which the interface is unreachable.</li> <li><b>route down (ifdn)</b>—Interface route for which the interface is unreachable.</li> <li><b>ignore (ignr)</b>—Ignore this route.</li> <li><b>interface (intf)</b>—Installed as a result of configuring an interface.</li> <li><b>permanent (perm)</b>—Routes installed by the kernel when the routing table is initialized.</li> <li><b>user</b>—Routes installed by the routing protocol process or as a result of the configuration.</li> </ul>	All levels
Route Reference (RtRef)	Number of routes to reference.	<b>detail extensive</b>
Flags	Route type flags: <ul style="list-style-type: none"> <li><b>none</b>—No flags are enabled.</li> <li><b>accounting</b>—Route has accounting enabled.</li> <li><b>cached</b>—Cache route.</li> <li><b>incoming-iface interface-number</b>—Check against incoming interface.</li> <li><b>prefix load balance</b>—Load balancing is enabled for this prefix.</li> <li><b>rt nh decoupled</b>—Route has been decoupled from the next hop to the destination.</li> <li><b>sent to PFE</b>—Route has been sent to the Packet Forwarding Engine.</li> <li><b>static</b>—Static route.</li> </ul>	<b>extensive</b>
Next hop	IP address of the next hop to the destination.	<b>detail extensive</b>

Table 34: show route forwarding-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Next hop Type (Type)	<p>Next-hop type. When the <b>detail</b> keyword is used, the next-hop type might be abbreviated (as indicated in parentheses):</p> <ul style="list-style-type: none"> <li>• <b>broadcast (bcst)</b>—Broadcast.</li> <li>• <b>deny</b>—Deny.</li> <li>• <b>discard (dscd)</b>—Discard.</li> <li>• <b>hold</b>—Next hop is waiting to be resolved into a unicast or multicast type.</li> <li>• <b>indexed (idxd)</b>—Indexed next hop.</li> <li>• <b>indirect (indr)</b>—Indirect next hop.</li> <li>• <b>local (locl)</b>—Local address on an interface.</li> <li>• <b>routed multicast (mcrst)</b>—Regular multicast next hop.</li> <li>• <b>multicast (mcst)</b>—Wire multicast next hop (limited to the LAN).</li> <li>• <b>multicast discard (mdsc)</b>—Multicast discard.</li> <li>• <b>multicast group (mgrp)</b>—Multicast group member.</li> <li>• <b>receive (rcv)</b>—Receive.</li> <li>• <b>reject (rjct)</b>—Discard. An ICMP unreachable message was sent.</li> <li>• <b>resolve (rslv)</b>—Resolving the next hop.</li> <li>• <b>unicast (ucst)</b>—Unicast.</li> <li>• <b>unilist (ulst)</b>—List of unicast next hops. A packet sent to this next hop goes to any next hop in the list.</li> </ul>	<b>detail extensive</b>
Index	Software index of the next hop that is used to route the traffic for a given prefix.	<b>detail extensive none</b>
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.	<b>extensive</b>
Reference (NhRef)	Number of routes that refer to this next hop.	<b>detail extensive none</b>
Next-hop interface (Netif)	Interface used to reach the next hop.	<b>detail extensive none</b>
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 <b>Balance</b> field description).	<b>extensive</b>
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.	<b>extensive</b>
RPF interface	List of interfaces from which the prefix can be accepted. Reverse path forwarding (RPF) information is displayed only when <b>rpf-check</b> is configured on the interface.	<b>extensive</b>

## 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                    locl  607  2
1.1.1.1/32       iddn  0 1.1.1.1                    locl  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                    locl  615  2
10.0.0.1/32      dest  0 10.0.0.1                    locl  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                    locl  611  2
10.1.1.1/32      iddn  0 10.1.1.1                    locl  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                locl  417  2
10.209.2.131/32  dest  0 10.209.2.131                locl  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                               locl  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          locl  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        locl  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         locl  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          locl  135   2
10.0.0.4/32      dest   0 10.0.0.4          locl  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 (Weights and Balances)

```

user@host> show route forwarding-table destination 3.4.2.1 extensive
Routing table: inet [Index 0]
Internet:

Destination: 3.4.2.1/32
Route type: user
Route reference: 0                               Route interface-index: 0

```

```

Flags: sent to PFE
Next-hop type: unilist           Index: 262143  Reference: 1
Nexthop: 4.4.4.4
Next-hop type: unicast          Index: 335     Reference: 2
Next-hop interface: so-1/1/0.0  Weight: 22    Balance: 3
Nexthop: 145.12.1.2
Next-hop type: unicast          Index: 337     Reference: 2
Next-hop interface: so-0/1/2.0  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                Route interface-index: 0
Flags: sent to PFE
Nexthop: 0:90:69:8e:b1:1b
Next-hop type: unicast           Index: 132     Reference: 4
Next-hop interface: fxp0.0

Destination: default
Route type: permanent
Route reference: 0                Route interface-index: 0
Flags: none
Next-hop type: reject            Index: 14      Reference: 1

Destination: 127.0.0.1/32
Route type: interface
Route reference: 0                Route interface-index: 0
Flags: sent to PFE
Nexthop: 127.0.0.1
Next-hop type: local              Index: 320     Reference: 1

...

Routing table: private1__inet [Index 1]
Internet:

Destination: default
Route type: permanent
Route reference: 0                Route interface-index: 0
Flags: sent to PFE
Next-hop type: reject            Index: 46      Reference: 1

Destination: 10.0.0.0/8
Route type: interface
Route reference: 0                Route interface-index: 3
Flags: sent to PFE
Next-hop type: resolve           Index: 136     Reference: 1
Next-hop interface: fxp1.0

...

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
Next-hop: 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          dymn  0
default          perm  0
fe-0/1/0.0       dymn  0
00:90:69:0c:20:1f/48 <<<<<Remote CE
                  dymn  0                  indr  351    4
                  Push 800000, Push 100002(top)

so-0/0/0.0
00:90:69:85:b0:1f/48 <<<<<Local CE
                  dymn  0                  ucst  354    2 fe-0/1/0.0

```

### show route forwarding-table vpls (Broadcast, unknown unicast, and multicast (BUM) hashing is enabled)

```

user@host> show route forwarding-table vpls
Routing table: green.vpls
VPLS:
Enabled protocols: BUM hashing
Destination      Type RtRef Next hop          Type Index NhRef Netif
default          perm  0
lsi.1048832      intf  0
                  4.4.3.2          indr 1048574 4
                  Push 262145      621    2

ge-3/0/0.0
00:19:e2:25:d0:01/48 user  0                  ucst  590    5 ge-2/3/9.0
0x30003/51       user  0                  comp  627    2
ge-2/3/9.0       intf  0                  ucst  590    5 ge-2/3/9.0
ge-3/1/3.0       intf  0                  ucst  619    4 ge-3/1/3.0
0x30002/51       user  0                  comp  600    2
0x30001/51       user  0                  comp  597    2

```

### show route forwarding-table vpls (Broadcast, unknown unicast, and multicast (BUM) hashing is enabled with MAC Statistics)

```

user@host> show route forwarding-table vpls
Routing table: green.vpls
VPLS:
Enabled protocols: BUM hashing, MAC Stats

```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		dscd	519	1	
1si.1048834	intf	0		indr	1048574	4	
			4.4.3.2	Push	262145	592	2
ge-3/0/0.0							
00:19:e2:25:d0:01/48	user	0		ucst	590	5	ge-2/3/9.0
0x30003/51	user	0		comp	630	2	
ge-2/3/9.0	intf	0		ucst	590	5	ge-2/3/9.0
ge-3/1/3.0	intf	0		ucst	591	4	ge-3/1/3.0
0x30002/51	user	0		comp	627	2	
0x30001/51	user	0		comp	624	2	

### 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 interface-index: 72
Route reference: 0	
Flags: sent to PFE	
Next-hop type: flood	Index: 289      Reference: 1
Next-hop type: unicast	Index: 291      Reference: 3
Next-hop interface: fe-0/1/3.0	
Next-hop type: unicast	Index: 290      Reference: 3
Next-hop interface: fe-0/1/2.0	

Destination: default

Route type: permanent	Route interface-index: 0
Route reference: 0	
Flags: none	
Next-hop type: discard	Index: 341      Reference: 1

Destination: fe-0/1/2.0

Route type: dynamic	Route interface-index: 69
Route reference: 0	
Flags: sent to PFE	
Next-hop type: flood	Index: 293      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: 291      Reference: 3
Next-hop interface: fe-0/1/3.0	

Destination: fe-0/1/3.0

Route type: dynamic	Route interface-index: 70
Route reference: 0	
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: 10:00:00:01:01:04/48
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/routing-instance-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           Type Index NhRef Netif
::/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 Nexthop      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     locl   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

<b>Syntax</b>	show route hidden <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4.
<b>Description</b>	Display only hidden route information. A hidden route is unusable, even if it is the best path.
<b>Options</b>	<p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b>.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Hidden Routes on page 1056</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show route hidden on page 970</a> <a href="#">show route hidden detail on page 971</a> <a href="#">show route hidden extensive on page 971</a> <a href="#">show route hidden terse on page 971</a>
<b>Output Fields</b>	For information about output fields, see the output field table for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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 971](#).

### 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.l3vpn.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

<b>List of Syntax</b>	<a href="#">Syntax on page 973</a> <a href="#">Syntax (EX Series Switches) on page 973</a>
<b>Syntax</b>	<pre>show route inactive-path &lt;brief   detail   extensive   terse&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route inactive-path &lt;brief   detail   extensive   terse&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display routes for destinations that have no active route. An inactive route is a route that was not selected as the best path.
<b>Options</b>	<p><b>none</b>—Display all inactive routes.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b>.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route inactive-path on page 973</a> <a href="#">show route inactive-path detail on page 974</a> <a href="#">show route inactive-path extensive on page 975</a> <a href="#">show route inactive-path terse on page 975</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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 974](#).

### 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           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 inactive-prefix

<b>List of Syntax</b>	<a href="#">Syntax on page 976</a> <a href="#">Syntax (EX Series Switches) on page 976</a>
<b>Syntax</b>	<pre>show route inactive-prefix &lt;brief   detail   extensive   terse&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route inactive-prefix &lt;brief   detail   extensive   terse&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	Display inactive route destinations in each routing table.
<b>Options</b>	<p><b>none</b>—Display all inactive route destination.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route inactive-prefix on page 976</a> <a href="#">show route inactive-prefix detail on page 976</a> <a href="#">show route inactive-prefix extensive on page 977</a> <a href="#">show route inactive-prefix terse on page 977</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> command.

## Sample Output

### show route inactive-prefix

```
user@host> show route inactive-prefix

inet.0: 14 destinations, 14 routes (13 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

127.0.0.1/32          [Direct/0] 00:04:54
> via lo0.0
```

### show route inactive-prefix detail

```
user@host> show route inactive-prefix detail

inet.0: 14 destinations, 14 routes (13 active, 0 holddown, 1 hidden)
```

```

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>
    Age: 4:51
    Task: IF
    AS path: I00:04:54
      > via lo0.0

```

### show route inactive-prefix extensive

The output for the **show route inactive-prefix extensive** command is identical to that of the **show route inactive-path detail** command. For sample output, see [show route inactive-prefix detail on page 976](#).

### show route inactive-prefix terse

```
user@host> show route inactive-prefix terse
```

```
inet.0: 18 destinations, 18 routes (17 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
127.0.0.1/32	D 0			>lo0.0	

## show route instance

---

<b>List of Syntax</b>	<a href="#">Syntax on page 978</a> <a href="#">Syntax (EX Series Switches and QFX Series) on page 978</a>
<b>Syntax</b>	<pre>show route instance &lt;brief   detail   summary&gt; &lt;instance-name&gt; &lt;logical-system (all   logical-system-name)&gt; &lt;operational&gt;</pre>
<b>Syntax (EX Series Switches and QFX Series)</b>	<pre>show route instance &lt;brief   detail   summary&gt; &lt;instance-name&gt; &lt;operational&gt;</pre>
<b>Release Information</b>	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.
<b>Description</b>	Display routing instance information.
<b>Options</b>	<p><b>none</b>—(Same as <b>brief</b>) Display standard information about all routing instances.</p> <p><b>brief   detail   summary</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b>. (These options are not available with the <b>operational</b> keyword.)</p> <p><b>instance-name</b>—(Optional) Display information for all routing instances whose name begins with this string (for example, <b>cust1</b>, <b>cust11</b>, and <b>cust111</b> are all displayed when you run the <b>show route instance cust1</b> command).</p> <p><b>logical-system (all   logical-system-name)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b>operational</b>—(Optional) Display operational routing instances.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Transporting IPv6 Traffic Across IPv4 Using Filter-Based Tunneling</i></li><li>• <i>Example: Configuring the Helper Capability Mode for OSPFv3 Graceful Restart</i></li></ul>
<b>List of Sample Output</b>	<a href="#">show route instance on page 980</a> <a href="#">show route instance detail (Graceful Restart Complete) on page 980</a> <a href="#">show route instance detail (Graceful Restart Incomplete) on page 982</a> <a href="#">show route instance detail (VPLS Routing Instance) on page 983</a> <a href="#">show route instance operational on page 984</a> <a href="#">show route instance summary on page 984</a>

**Output Fields** Table 35 on page 979 lists the output fields for the **show route instance** command. Output fields are listed in the approximate order in which they appear.

**Table 35: 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
Operational Routing Instances	( <b>operational</b> keyword only) Names of all operational routing instances.	—
Type	Type of routing instance: <b>forwarding</b> , <b>l2vpn</b> , <b>no-forwarding</b> , <b>vpls</b> , <b>virtual-router</b> , or <b>vrf</b> .	All levels
State	State of the routing instance: <b>active</b> or <b>inactive</b> .	<b>brief detail none</b>
Interfaces	Name of interfaces belonging to this routing instance.	<b>brief detail none</b>
Restart State	Status of graceful restart for this instance: <b>Pending</b> or <b>Complete</b> .	<b>detail</b>
Path selection timeout	Maximum amount of time, in seconds, remaining until graceful restart is declared complete. The default is <b>300</b> .	<b>detail</b>
Tables	Tables (and number of routes) associated with this routing instance.	<b>brief detail none</b>
Route-distinguisher	Unique route distinguisher associated with this routing instance.	<b>detail</b>
Vrf-import	VPN routing and forwarding instance import policy name.	<b>detail</b>
Vrf-export	VPN routing and forwarding instance export policy name.	<b>detail</b>
Vrf-import-target	VPN routing and forwarding instance import target community name.	<b>detail</b>
Vrf-export-target	VPN routing and forwarding instance export target community name.	<b>detail</b>
Fast-reroute-priority	Fast reroute priority setting for a VPLS routing instance: <b>high</b> , <b>medium</b> , or <b>low</b> . The default is <b>low</b> .	<b>detail</b>
Restart State	Restart state: <ul style="list-style-type: none"> <li><b>Pending;protocol-name</b>—List of protocols that have not yet completed graceful restart for this routing table.</li> <li><b>Complete</b>—All protocols have restarted for this routing table.</li> </ul>	<b>detail</b>
Primary rib	Primary table for this routing instance.	<b>brief none summary</b>
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

```

```

1si.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
		l2vpn.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

<b>List of Syntax</b>	<a href="#">Syntax on page 986</a> <a href="#">Syntax (EX Series Switches) on page 986</a>
<b>Syntax</b>	<b>show route next-hop</b> <i>next-hop</i> <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	<b>show route next-hop</b> <i>next-hop</i> <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display the entries in the routing table that are being sent to the specified next-hop address.
<b>Options</b>	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.  <b><i>next-hop</i></b> —Next-hop address.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route next-hop on page 986</a> <a href="#">show route next-hop detail on page 987</a> <a href="#">show route next-hop extensive on page 989</a> <a href="#">show route next-hop terse on page 990</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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-kerne1 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-kerne1 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 no-community

<b>List of Syntax</b>	<a href="#">Syntax on page 992</a> <a href="#">Syntax (EX Series Switches) on page 992</a>
<b>Syntax</b>	show route no-community <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	show route no-community <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display the route entries in each routing table that are not associated with any community.
<b>Options</b>	<p><b>none</b>—(Same as <b>brief</b>) Display the route entries in each routing table that are not associated with any community.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route no-community on page 992</a> <a href="#">show route no-community detail on page 993</a> <a href="#">show route no-community extensive on page 993</a> <a href="#">show route no-community terse on page 994</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> command.

## Sample Output

### show route no-community

```

user@host> show route no-community
inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.0.0/16      *[Static/5] 00:36:27
> to 192.168.71.254 via fxp0.0
10.209.0.0/16    *[Static/5] 00:36:27
> to 192.168.71.254 via fxp0.0
10.255.71.52/32  *[Direct/0] 00:36:27
> via lo0.0
10.255.71.63/32  *[OSPF/10] 00:04:39, metric 1
> to 35.1.1.2 via ge-3/1/0.0
10.255.71.64/32  *[OSPF/10] 00:00:08, metric 2

```

```

> to 35.1.1.2 via ge-3/1/0.0
10.255.71.240/32  * [OSPF/10] 00:05:04, metric 2
                  via so-0/1/2.0
> via so-0/3/2.0
10.255.71.241/32  * [OSPF/10] 00:05:14, metric 1
> via so-0/1/2.0
10.255.71.242/32  * [OSPF/10] 00:05:19, metric 1
> via so-0/3/2.0
12.1.1.0/24       * [OSPF/10] 00:05:14, metric 2
> via so-0/3/2.0
14.1.1.0/24       * [OSPF/10] 00:00:08, 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] 00:05:14, metric 2
> via so-0/1/2.0
.....

```

### show route no-community detail

```

user@host> show route no-community detail

inet.0: 28 destinations, 30 routes (27 active, 0 holddown, 1 hidden)
10.10.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Age: 38:08
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

10.209.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 22
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Age: 38:08
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

....

```

### show route no-community extensive

```

user@host> show route no-community 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: 2:03:33
    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: 2:03:33
    Task: RT
    Announcement bits (1): 0-KRT
    AS path: I

```

### show route no-community terse

```
user@host> show route no-community 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
*	10.10.0.0/16	S	5			>192.168.71.254	
*	10.209.0.0/16	S	5			>192.168.71.254	
*	10.255.71.52/32	D	0			>lo0.0	
*	10.255.71.63/32	O	10	1		>35.1.1.2	
*	10.255.71.64/32	O	10	2		>35.1.1.2	
*	10.255.71.240/32	O	10	2		so-0/1/2.0	
						>so-0/3/2.0	
*	10.255.71.241/32	O	10	1		>so-0/1/2.0	
*	10.255.71.242/32	O	10	1		>so-0/3/2.0	
*	12.1.1.0/24	O	10	2		>so-0/3/2.0	
*	14.1.1.0/24	O	10	3		>35.1.1.2	
						so-0/1/2.0	
						so-0/3/2.0	
*	16.1.1.0/24	O	10	2		>so-0/1/2.0	

...

## show route output

<b>List of Syntax</b>	<a href="#">Syntax on page 995</a> <a href="#">Syntax (EX Series Switches) on page 995</a>
<b>Syntax</b>	<pre>show route output (address <i>ip-address</i>   interface <i>interface-name</i>) &lt;brief   detail   extensive   terse&gt; &lt;logical-system (all   <i>logical-system-name</i>)&gt;</pre>
<b>Syntax (EX Series Switches)</b>	<pre>show route output (address <i>ip-address</i>   interface <i>interface-name</i>) &lt;brief   detail   extensive   terse&gt;</pre>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p>
<b>Description</b>	<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 <b>show route advertising-protocol bgp</b> and <b>show route receive-protocol bgp</b> commands instead.</p>
<b>Options</b>	<p><b>address <i>ip-address</i></b>—Display entries in the routing table that are to be sent out the interface with the specified IP address.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b>.</p> <p><b>interface <i>interface-name</i></b>—Display entries in the routing table that are to be sent out the interface with the specified name.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route output address on page 996</a> <a href="#">show route output address detail on page 996</a> <a href="#">show route output address extensive on page 997</a> <a href="#">show route output address terse on page 997</a> <a href="#">show route output interface on page 997</a> <a href="#">show route output interface detail on page 998</a> <a href="#">show route output interface extensive on page 998</a> <a href="#">show route output interface terse on page 998</a>
<b>Output Fields</b>	<p>For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> command.</p>

## 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 996](#).

### 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                >so-0/1/2.0
                   O  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 998](#).

### 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

---

<b>List of Syntax</b>	<a href="#">Syntax on page 1000</a> <a href="#">Syntax (EX Series Switches) on page 1000</a>
<b>Syntax</b>	<code>show route protocol <i>protocol</i></code> <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	<code>show route protocol <i>protocol</i></code> <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. <b>ospf2</b> and <b>ospf3</b> options introduced in Junos OS Release 9.2. <b>ospf2</b> and <b>ospf3</b> options introduced in Junos OS Release 9.2 for EX Series switches. <b>flow</b> option introduced in Junos OS Release 10.0. <b>flow</b> option introduced in Junos OS Release 10.0 for EX Series switches.
<b>Description</b>	Display the route entries in the routing table that were learned from a particular protocol.
<b>Options</b>	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to <b>brief</b> .  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.  <b><i>protocol</i></b> —Protocol from which the route was learned: <ul style="list-style-type: none"><li>• <b>access</b>—Access route for use by DHCP application</li><li>• <b>access-internal</b>—Access-internal route for use by DHCP application</li><li>• <b>aggregate</b>—Locally generated aggregate route</li><li>• <b>arp</b>—Route learned through the Address Resolution Protocol</li><li>• <b>atmvpn</b>—Asynchronous Transfer Mode virtual private network</li><li>• <b>bgp</b>—Border Gateway Protocol</li><li>• <b>ccc</b>—Circuit cross-connect</li><li>• <b>direct</b>—Directly connected route</li><li>• <b>dvmrp</b>—Distance Vector Multicast Routing Protocol</li><li>• <b>esis</b>—End System-to-Intermediate System</li><li>• <b>flow</b>—Locally defined flow-specification route</li><li>• <b>frr</b>—Precomputed protection route or backup route used when a link goes down</li><li>• <b>isis</b>—Intermediate System-to-Intermediate System</li><li>• <b>ldp</b>—Label Distribution Protocol</li><li>• <b>l2circuit</b>—Layer 2 circuit</li></ul>

- **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.

<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route protocol access on page 1002</a> <a href="#">show route protocol access-internal extensive on page 1002</a> <a href="#">show route protocol arp on page 1002</a> <a href="#">show route protocol bgp on page 1003</a> <a href="#">show route protocol bgp detail on page 1003</a> <a href="#">show route protocol bgp extensive on page 1003</a> <a href="#">show route protocol bgp terse on page 1004</a> <a href="#">show route protocol direct on page 1004</a> <a href="#">show route protocol frr on page 1005</a> <a href="#">show route protocol l2circuit detail on page 1005</a> <a href="#">show route protocol l2vpn extensive on page 1006</a> <a href="#">show route protocol ldp on page 1007</a> <a href="#">show route protocol ldp extensive on page 1007</a> <a href="#">show route protocol ospf (Layer 3 VPN) on page 1008</a> <a href="#">show route protocol ospf detail on page 1009</a> <a href="#">show route protocol rip on page 1009</a> <a href="#">show route protocol rip detail on page 1009</a>

[show route protocol ripng table inet6 on page 1010](#)

[show route protocol static detail on page 1010](#)

**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-kernel 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: l2 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-kernel 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-kernel 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

<b>List of Syntax</b>	<a href="#">Syntax on page 1012</a> <a href="#">Syntax (EX Series Switches) on page 1012</a>
<b>Syntax</b>	show route receive-protocol <i>protocol neighbor-address</i> <brief   detail   extensive   terse> <logical-system (all   <i>logical-system-name</i> )>
<b>Syntax (EX Series Switches)</b>	show route receive-protocol <i>protocol neighbor-address</i> <brief   detail   extensive   terse>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Display the routing information as it was received through a particular neighbor using a particular dynamic routing protocol.
<b>Options</b>	<p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output.</p> <p><b>logical-system (all   <i>logical-system-name</i>)</b>—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><b><i>protocol neighbor-address</i></b>—Protocol transmitting the route (<b>bgp</b>, <b>dvmrp</b>, <b>msdp</b>, <b>pim</b>, <b>rip</b>, or <b>ripng</b>) and address of the neighboring router from which the route entry was received.</p>
<b>Additional Information</b>	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.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show route receive-protocol bgp on page 1015</a> <a href="#">show route receive-protocol bgp extensive on page 1015</a> <a href="#">show route receive-protocol bgp table extensive on page 1015</a> <a href="#">show route receive-protocol bgp logical-system extensive on page 1016</a> <a href="#">show route receive-protocol bgp detail (Layer 2 VPN) on page 1017</a> <a href="#">show route receive-protocol bgp extensive (Layer 2 VPN) on page 1017</a> <a href="#">show route receive-protocol bgp (Layer 3 VPN) on page 1018</a> <a href="#">show route receive-protocol bgp detail (Layer 3 VPN) on page 1018</a> <a href="#">show route receive-protocol bgp extensive (Layer 3 VPN) on page 1019</a>
<b>Output Fields</b>	<a href="#">Table 36 on page 1012</a> describes the output fields for the <b>show route receive-protocol</b> command. Output fields are listed in the approximate order in which they appear.

**Table 36: 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

Table 36: show route receive-protocol Output Fields (*continued*)

Field Name	Field Description	Level of Output
<i>number destinations</i>	Number of destinations for which there are routes in the routing table.	All levels
<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"> <li>• <b>active</b></li> <li>• <b>holddown</b> (routes that are in pending state before being declared inactive)</li> <li>• <b>hidden</b> (routes that are not used because of a routing policy)</li> </ul>	All levels
Prefix	Destination prefix.	none <b>brief</b>
MED	Multiple exit discriminator value included in the route.	none <b>brief</b>
<i>destination-prefix (entry, announced)</i>	Destination prefix. The <b>entry</b> value is the number of routes for this destination, and the <b>announced</b> value is the number of routes being announced for this destination.	<b>detail extensive</b>
Route Distinguisher	64-bit prefix added to IP subnets to make them unique.	<b>detail extensive</b>
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.	<b>detail extensive</b>
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.	<b>detail extensive</b>
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 36: 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"> <li>• <b>I</b>—IGP.</li> <li>• <b>E</b>—EGP.</li> <li>• <b>?</b>—Incomplete; typically, the AS path was aggregated.</li> </ul> <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> <li>• <b>[ ]</b>—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.</li> <li>• <b>[ ]</b>—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.</li> <li>• <b>{ }</b>—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.</li> <li>• <b>( )</b>—Parentheses enclose a confederation.</li> <li>• <b>( [ ] )</b>—Parentheses and brackets enclose a confederation set.</li> </ul> <p><b>NOTE:</b> 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 <a href="#">show route detail</a> 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 <b>Attrset</b> attribute at the originating routing device.	detail extensive
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).	detail extensive
control flags	Control flags: <b>none</b> or <b>Site Down</b> .	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 logical-system extensive**

```
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)
  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 extensive (Layer 2 VPN)**

```

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
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 extensive (Layer 3 VPN)

```

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
  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

---

List of Syntax	<a href="#">Syntax on page 1020</a> <a href="#">Syntax (EX Series Switches) on page 1020</a>
Syntax	<code>show route table <i>routing-table-name</i></code> <code>&lt;brief   detail   extensive   terse&gt;</code> <code>&lt;logical-system (all   <i>logical-system-name</i>)&gt;</code>
Syntax (EX Series Switches)	<code>show route table <i>routing-table-name</i></code> <code>&lt;brief   detail   extensive   terse&gt;</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 entries in a particular routing table.
Options	<b>brief   detail   extensive   terse</b> —(Optional) Display the specified level of output.  <b>logical-system (all   <i>logical-system-name</i>)</b> —(Optional) Perform this operation on all logical systems or on a particular logical system.  <b><i>routing-table-name</i></b> —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 <b>show route table inet</b> command).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"><li>• <a href="#">show route summary</a></li></ul>
List of Sample Output	<a href="#">show route table bgp.l2.vpn on page 1021</a> <a href="#">show route table bgp.l3vpn.0 on page 1021</a> <a href="#">show route table bgp.l3vpn.0 detail on page 1021</a> <a href="#">show route table bgp.rtarget.0 (When Proxy BGP Route Target Filtering Is Configured) on page 1023</a> <a href="#">show route table inet.0 on page 1023</a> <a href="#">show route table inet6.0 on page 1023</a> <a href="#">show route table inet6.3 on page 1024</a> <a href="#">show route table inetflow detail on page 1024</a> <a href="#">show route table l2circuit.0 on page 1024</a> <a href="#">show route table mpls on page 1025</a> <a href="#">show route table mpls extensive on page 1025</a> <a href="#">show route table mpls.0 on page 1025</a> <a href="#">show route table mpls.0 detail (PTX Series) on page 1026</a> <a href="#">show route table mpls.0 extensive (PTX Series) on page 1026</a> <a href="#">show route table mpls.0 (RSVP Route—Transit LSP) on page 1027</a> <a href="#">show route table vpls_1 detail on page 1027</a> <a href="#">show route table vpn-a on page 1028</a>

[show route table vpn-a.mdt.0 on page 1028](#)  
[show route table VPN-A detail on page 1028](#)  
[show route table VPN-AB.inet.0 on page 1029](#)  
[show route table VPN\\_blue.mvpn-inet6.0 on page 1029](#)  
[show route table vrf1.mvpn.0 extensive on page 1030](#)  
[show route table inetflow detail on page 1030](#)

**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.l2.vpn

```

user@host> show route table bgp.l2.vpn
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 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 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 Target Filtering Is Configured)

```

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

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

```

```

100004          via so-0/1/3.0, Swap 100002
                *[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 detail (PTX Series)

```

user@host> show route table mpls.0 detail
ge-0/0/2.600 (1 entry, 1 announced)
  *L2VPN Preference: 7
    Next hop type: Indirect
    Address: 0x9438f34
    Next-hop reference count: 2
    Next hop type: Router, Next hop index: 567
    Next hop: 3.0.0.1 via ge-0/0/1.0, selected
    Label operation: Push 299808
    Label TTL action: prop-ttl
    Load balance label: Label 299808:None;
    Session Id: 0x1
    Protocol next hop: 10.255.255.1
    Label operation: Push 299872 Offset: 252
    Label TTL action: no-prop-ttl
    Load balance label: Label 299872:Flow label PUSH;
    Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
    Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
    State: <Active Int>
    Age: 21 Metric2: 1
    Validation State: unverified
    Task: Common L2 VC
    Announcement bits (2): 0-KRT 2-Common L2 VC
    AS path: I

```

#### show route table mpls.0 extensive (PTX Series)

```

user@host> show route table mpls.0 extensive
ge-0/0/2.600 (1 entry, 1 announced)
TSI:
KRT in-kernel ge-0/0/2.600.0 /32 -> {composite(570)}
  *L2VPN Preference: 7
    Next hop type: Indirect
    Address: 0x9438f34
    Next-hop reference count: 2
    Next hop type: Router, Next hop index: 567
    Next hop: 3.0.0.1 via ge-0/0/1.0, selected
    Label operation: Push 299808
    Label TTL action: prop-ttl
    Load balance label: Label 299808:None;
    Session Id: 0x1
    Protocol next hop: 10.255.255.1
    Label operation: Push 299872 Offset: 252
    Label TTL action: no-prop-ttl
    Load balance label: Label 299872:Flow label PUSH;
    Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
    Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
    State: <Active Int>

```

```

Age: 47          Metric2: 1
Validation State: unverified
Task: Common L2 VC
Announcement bits (2): 0-KRT 2-Common L2 VC
AS path: I
Composite next hops: 1
  Protocol next hop: 10.255.255.1 Metric: 1
  Label operation: Push 299872 Offset: 252
  Label TTL action: no-prop-ttl
  Load balance label: Label 299872:Flow label PUSH;
  Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
  Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
  Indirect path forwarding next hops: 1
    Next hop type: Router
    Next hop: 3.0.0.1 via ge-0/0/1.0
    Session Id: 0x1
  10.255.255.1/32 Originating RIB: inet.3
    Metric: 1                      Node path count: 1
    Forwarding nexthops: 1
      Nexthop: 3.0.0.1 via ge-0/0/1.0

```

### 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.12vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both
192.168.16.1:1:1:1/96
    *[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.13vpn.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 vrf1.mvpn.0 extensive

```

user@host> show route table vrf1.mvpn.0 extensive
1:10.255.50.77:1:10.255.50.77/240 (1 entry, 1 announced)
    *MVPN    Preference: 70
              PMSI: Flags 0x0: Label 0: RSVP-TE:
Session_13[10.255.50.77:0:25624:10.255.50.77]
    Next hop type: Indirect
    Address: 0xbb2c944
    Next-hop reference count: 360
    Protocol next hop: 10.255.50.77
    Indirect next hop: 0x0 - INH Session ID: 0x0
    State: <Active Int Ext>
    Age: 53:03      Metric2: 1
    Validation State: unverified
    Task: mvpn global task
    Announcement bits (3): 0-PIM.vrf1 1-mvpn global task 2-rt-export

    AS path: I

```

### 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

user@host> show route table red extensive
red.inet.0: 364481 destinations, 714087 routes (364480 active, 48448 holddown, 1
hidden)
22.0.0.0/32 (3 entries, 1 announced)
    State: <OnList CalcForwarding>
TSI:
KRT in-kernel 22.0.0.0/32 -> {composite(1048575)} Page 0 idx 1 Type 1 val 0x934342c

    Nexthop: Self
    AS path: [2] I
    Communities: target:2:1
Path 22.0.0.0 from 2.3.0.0 Vector len 4. Val: 1
    @BGP Preference: 170/-1
    Route Distinguisher: 2:1
    Next hop type: Indirect
    Address: 0x258059e4
    Next-hop reference count: 2
    Source: 2.2.0.0
    Next hop type: Router
    Next hop: 10.1.1.1 via ge-1/1/9.0, selected
    Label operation: Push 707633
    Label TTL action: prop-ttl
    Session Id: 0x17d8
    Protocol next hop: 2.2.0.0
    Push 16
    Composite next hop: 0x25805988 - INH Session ID: 0x193c
    Indirect next hop: 0x23eea900 - INH Session ID: 0x193c
    State: <Secondary Active Int Ext ProtectionPath ProtectionCand>
    Local AS: 2 Peer AS: 2
    Age: 23 Metric2: 35
    Validation State: unverified
    Task: BGP_2.2.2.0.0+34549
    AS path: I
    Communities: target:2:1
    Import Accepted
    VPN Label: 16
    Localpref: 0
    Router ID: 2.2.0.0
    Primary Routing Table bgp.13vpn.0
    Composite next hops: 1
        Protocol next hop: 2.2.0.0 Metric: 35
        Push 16
        Composite next hop: 0x25805988 - INH Session ID: 0x193c
        Indirect next hop: 0x23eea900 - INH Session ID: 0x193c
        Indirect path forwarding next hops: 1
            Next hop type: Router
            Next hop: 10.1.1.1 via ge-1/1/9.0

```

```

                Session Id: 0x17d8
                2.2.0.0/32 Originating RIB: inet.3
                Metric: 35                      Node path count: 1
                Forwarding nexthops: 1
                Nexthop: 10.1.1.1 via ge-1/1/9.0
BGP  Preference: 170/-1
      Route Distinguisher: 2:1
      Next hop type: Indirect
      Address: 0x9347028
      Next-hop reference count: 3
      Source: 2.3.0.0
      Next hop type: Router, Next hop index: 702
      Next hop: 10.1.4.2 via ge-1/0/0.0, selected
      Label operation: Push 634278
      Label TTL action: prop-ttl
      Session Id: 0x17d9
      Protocol next hop: 2.3.0.0
      Push 16
      Composite next hop: 0x93463a0 1048575 INH Session ID: 0x17da
      Indirect next hop: 0x91e8800 1048574 INH Session ID: 0x17da
      State: <Secondary NotBest Int Ext ProtectionPath ProtectionCand>

      Inactive reason: Not Best in its group - IGP metric
      Local AS:      2 Peer AS:      2
      Age: 3:34      Metric2: 70
      Validation State: unverified
      Task: BGP_2.2.3.0.0+32805
      Announcement bits (2): 0-KRT 1-BGP_RT_Background
      AS path: I
      Communities: target:2:1
      Import Accepted
      VPN Label: 16
      Localpref: 0
      Router ID: 2.3.0.0
      Primary Routing Table bgp.13vpn.0
      Composite next hops: 1
        Protocol next hop: 2.3.0.0 Metric: 70
        Push 16
        Composite next hop: 0x93463a0 1048575 INH Session ID:
0x17da      Indirect next hop: 0x91e8800 1048574 INH Session ID:
0x17da      Indirect path forwarding next hops: 1
              Next hop type: Router
              Next hop: 10.1.4.2 via ge-1/0/0.0
              Session Id: 0x17d9
              2.3.0.0/32 Originating RIB: inet.3
              Metric: 70                      Node path count: 1
              Forwarding nexthops: 1
              Nexthop: 10.1.4.2 via ge-1/0/0.0
#Multipath Preference: 255
      Next hop type: Indirect
      Address: 0x24afca30
      Next-hop reference count: 1
      Next hop type: Router
      Next hop: 10.1.1.1 via ge-1/1/9.0, selected
      Label operation: Push 707633
      Label TTL action: prop-ttl
      Session Id: 0x17d8
      Next hop type: Router, Next hop index: 702
      Next hop: 10.1.4.2 via ge-1/0/0.0

```

```

Label operation: Push 634278
Label TTL action: prop-ttl
Session Id: 0x17d9
Protocol next hop: 2.2.0.0
Push 16
Composite next hop: 0x25805988 - INH Session ID: 0x193c
Indirect next hop: 0x23eea900 - INH Session ID: 0x193c Weight 0x1

Protocol next hop: 2.3.0.0
Push 16
Composite next hop: 0x93463a0 1048575 INH Session ID: 0x17da
Indirect next hop: 0x91e8800 1048574 INH Session ID: 0x17da Weight

0x4000
State: <ForwardingOnly Int Ext>
Inactive reason: Forwarding use only
Age: 23          Metric2: 35
Validation State: unverified
Task: RT
AS path: I
Communities: target:2:1

```

## show route terse

**List of Syntax** [Syntax on page 1034](#)  
[Syntax \(EX Series Switches\) on page 1034](#)

**Syntax** show route terse  
 <logical-system (all | *logical-system-name*)>

**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.



**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.

To display the metric1 and metric2 value of a BGP route, use the [show route extensive](#) command.

**Options** **none**—Display a high-level summary of the routes in the routing table.

**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 terse on page 1036](#)

**Output Fields** [Table 37 on page 1034](#) describes the output fields for the **show route terse** command. Output fields are listed in the approximate order in which they appear.

**Table 37: 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"> <li><b>active</b> (routes that are active)</li> <li><b>holddown</b> (routes that are in the pending state before being declared inactive)</li> <li><b>hidden</b> (routes that are not used because of a routing policy)</li> </ul>

Table 37: show route terse Output Fields (*continued*)

Field Name	Field Description
<i>route key</i>	<p>Key for the state of the route:</p> <ul style="list-style-type: none"> <li>• <b>+</b>—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table.</li> <li>• <b>-</b>—A hyphen indicates the last active route.</li> <li>• <b>*</b>—An asterisk indicates that the route is both the active and the last active route. An asterisk before a <b>to</b> line indicates the best subpath to the route.</li> </ul>
<b>A</b>	Active route. An asterisk (*) indicates this is the active route.
<b>V</b>	<p>Validation status of the route:</p> <ul style="list-style-type: none"> <li>• <b>?</b>—Not evaluated. Indicates that the route was not learned through BGP.</li> <li>• <b>I</b>—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.</li> <li>• <b>N</b>—Unknown. Indicates that the prefix is not among the prefixes or prefix ranges in the database.</li> <li>• <b>V</b>—Valid. Indicates that the prefix and autonomous system pair are found in the database.</li> </ul>
Destination	Destination of the route.
<b>P</b>	<p>Protocol through which the route was learned:</p> <ul style="list-style-type: none"> <li>• <b>A</b>—Aggregate</li> <li>• <b>B</b>—BGP</li> <li>• <b>C</b>—CCC</li> <li>• <b>D</b>—Direct</li> <li>• <b>G</b>—GMPLS</li> <li>• <b>I</b>—IS-IS</li> <li>• <b>L</b>—L2CKT, L2VPN, LDP, Local</li> <li>• <b>K</b>—Kernel</li> <li>• <b>M</b>—MPLS, MSDP</li> <li>• <b>O</b>—OSPF</li> <li>• <b>P</b>—PIM</li> <li>• <b>R</b>—RIP, RIPng</li> <li>• <b>S</b>—Static</li> <li>• <b>T</b>—Tunnel</li> </ul>
<b>Prf</b>	<p>Preference value of the route. In every routing metric except for the BGP <b>LocalPref</b> attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the <b>LocalPref</b> value in the <b>Preference2</b> field. For example, if the <b>LocalPref</b> value for Route 1 is 100, the <b>Preference2</b> value is -101. If the <b>LocalPref</b> value for Route 2 is 155, the <b>Preference2</b> value is -156. Route 2 is preferred because it has a higher <b>LocalPref</b> value and a lower <b>Preference2</b> value.</p>
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 37: 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"> <li>I—IGP.</li> <li>E—EGP.</li> <li>?—Incomplete; typically, the AS path was aggregated.</li> </ul>

## 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       0 10      1          >10.0.0.2      I
?                               B 170      100          >10.0.0.2      I
  unverified
* ? 1.1.1.1/32       D 0          >10.0.0.2      200 I
* V 2.2.0.2/32       B 170     110          >10.0.0.2
  valid
* ? 10.0.0.0/30      D 0          >1t-1/2/0.1    I
?                               B 170     100          >10.0.0.2
  unverified
* ? 10.0.0.1/32      L 0          Local          I
* ? 10.0.0.4/30      B 170     100          >10.0.0.2
  unverified
* ? 10.0.0.8/30      B 170     100          >10.0.0.2
  unverified
* I 172.16.1.1/32     B 170      90          >10.0.0.2      200 I
  invalid
* N 192.168.2.3/32   B 170     100          >10.0.0.2      200 I
  unknown
* ? 224.0.0.5/32     O 10      1          MultiRecv

```

## show security keychain

<b>Syntax</b>	show security keychain <brief   detail>
<b>Release Information</b>	Command introduced in Junos OS Release 11.2.
<b>Description</b>	Display information about authentication keychains configured for the Border Gateway Protocol (BGP), the Label Distribution Protocol (LDP) routing protocols, the Bidirectional Forwarding Detection (BFD) protocol, and the Intermediate System-to-Intermediate System (IS-IS) protocol.
<b>Options</b>	<b>none</b> —Display information about authentication keychains.  <b>brief   detail</b> —(Optional) Display the specified level of output.
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<a href="#">show security keychain brief on page 1039</a> <a href="#">show security keychain detail on page 1039</a>
<b>Output Fields</b>	<a href="#">Table 38 on page 1037</a> describes the output fields for the <b>show security keychain</b> command. Output fields are listed in the approximate order in which they appear.

**Table 38: show security keychain Output Fields**

Field Name	Field Description	Level of Output
<b>keychain</b>	The name of the keychain in operation.	All levels
<b>Active-ID Send</b>	Number of routing protocols packets sent with the active key.	All levels
<b>Active-ID Receive</b>	Number of routing protocols packets received with the active key.	All levels
<b>Next-ID Send</b>	Number of routing protocols packets sent with the next key.	All levels
<b>Next-ID Receive</b>	Number of routing protocols packets received with the next key.	All levels
<b>Transition</b>	Amount of time until the current key will be replaced with the next key in the keychain.	All levels
<b>Tolerance</b>	Configured clock-skew tolerance, in seconds, for accepting keys for a key chain.	All levels
<b>Id</b>	Identification number configured for the current key.	<b>detail</b>

Table 38: show security keychain Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Algorithm</b>	Authentication algorithm configured for the current key.	<b>detail</b>
<b>State</b>	<p>State of the current key.</p> <p>The value can be:</p> <ul style="list-style-type: none"> <li>• <b>receive</b></li> <li>• <b>send</b></li> <li>• <b>send-receive</b></li> </ul> <p>For the active key, the <b>State</b> can be <b>send-receive</b>, <b>send</b>, or <b>receive</b>. For keys that have a future start time, the <b>State</b> is <b>inactive</b>. Compare the <b>State</b> field to the <b>Mode</b> field.</p>	<b>detail</b>
<b>Option</b>	<p>For IS-IS only, the option determines how Junos OS encodes the message authentication code in routing protocol packets.</p> <p>The values can be:</p> <ul style="list-style-type: none"> <li>• <b>basic</b>—Based on RFC 5304.</li> <li>• <b>isis-enhanced</b>—Based on RFC 5310.</li> </ul> <p>The default value is <b>basic</b>. When you configure the <b>isis-enhanced</b> option, Junos OS sends RFC 5310-encoded routing protocol packets and accepts both RFC 5304-encoded and RFC 5310-encoded routing protocol packets that are received from other devices.</p> <p>When you configure <b>basic</b> (or do not include the <b>options</b> statement in the key configuration) Junos OS sends and receives RFC 5304-encoded routing protocols packets, and drops 5310-encoded routing protocol packets that are received from other devices.</p> <p>Because this setting is for IS-IS only, the TCP and the BFD protocol ignore the encoding option configured in the key.</p>	<b>detail</b>
<b>Start-time</b>	Time that the current key became active.	<b>detail</b>

Table 38: show security keychain Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Mode</b>	<p>Mode of each key (Informational only.)</p> <p>The value can be</p> <ul style="list-style-type: none"> <li>• <b>receive</b></li> <li>• <b>send</b></li> <li>• <b>send-receive</b></li> </ul> <p>The mode of the key is based on the configuration. Suppose you configure two keys, one with a start-time of today and the other with a start-time of next week. For both keys, the <b>Mode</b> can be <b>send-receive</b>, <b>send</b>, or <b>receive</b>, regardless of the configured start-time. Compare the <b>Mode</b> field to the <b>State</b> field.</p>	<b>detail</b>

## Sample Output

### show security keychain brief

```

user@host> show security keychain brief
keychain              Active-ID      Next-ID      Transition  Tolerance
                      Send  Receive    Send  Receive
hakr                   3     3           1     1         1d 23:58    3600

```

### show security keychain detail

```

user@host> show security keychain detail
keychain              Active-ID      Next-ID      Transition  Tolerance
                      Send  Receive    Send  Receive
hakr                   3     3           1     1         1d 23:58    3600
  Id 3, Algorithm hmac-md5, State send-receive, Option basic
  Start-time Wed Aug 11 16:28:00 2010, Mode send-receive
  Id 1, Algorithm hmac-md5, State inactive, Option basic
  Start-time Fri Aug 20 11:30:57 2010, Mode send-receive

```

## show validation database

<b>Syntax</b>	<pre>show validation database &lt;brief   detail&gt; &lt;instance <i>instance-name</i>&gt; &lt;logical-system <i>logical-system-name</i>&gt; &lt;mismatch&gt; &lt;origin-autonomous-system <i>as-number</i>&gt; &lt;record <i>ip-prefix</i>&gt; &lt;session <i>ip-address</i>&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Display information about the route validation database when resource public key infrastructure (RPKI) BGP route validation is configured. You can query all route validation records that match a given prefix or origin-autonomous-system. In addition, you can filter the output by a specific RPKI cache session.
<b>Options</b>	<p><b>none</b>—Display all route validation database entries.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information about route validation database entries for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p> <p><b>mismatch</b>—(Optional) Filter the output by mismatched origin autonomous systems.</p> <p><b>origin-autonomous-system <i>as-number</i></b>—(Optional) Filter the output by mismatched origin autonomous systems. The <b>mismatch</b> qualifier is useful for finding conflicting origin-autonomous-system information between RPKI caches. Mismatches might occur during cache reconfiguration.</p> <p><b>record <i>ip-prefix</i></b>—(Optional) Filter the output by route validation records that match a given prefix.</p> <p><b>session <i>ip-address</i></b>—(Optional) Filter the output by a specific RPKI cache session.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show validation database on page 1041</a>
<b>Output Fields</b>	<a href="#">Table 39 on page 1041</a> describes the output fields for the <b>show validation database</b> command. Output fields are listed in the approximate order in which they appear.

Table 39: show validation database Output Fields

Field Name	Field Description	Level of Output
Prefix	Route validation (RV) record prefix.  RV records are received from the cache server and can also be configured statically at the <code>[edit routing-options validation static]</code> hierarchy level .	All levels
Origin-AS	Legitimate originator autonomous system (AS).	All levels
Session	IP address of the RPKI cache server.	All levels
State	State of the route validation records. The state can be <b>valid</b> , <b>invalid</b> or <b>unknown</b> .	All levels
Mismatch	Conflicting origin-autonomous-system information between RPKI caches when nonstop active routing (NSR) is configured.	All levels
IPv4 records	Number of IPv4 route validation records.	All levels
IPv6 records	Number of IPv6 route validation records.	All levels

## Sample Output

### show validation database

```

user@host> show validation database
RV database for instance master

    Prefix                Origin-AS Session      State  Mismatch
    1.0.1.0/24-32          1 10.0.77.1    valid
    1.0.2.0/24-32          2 10.0.77.1    valid
    1.0.3.0/24-32          3 10.0.77.1    valid
    1.0.4.0/24-32          4 10.0.77.1    valid
    1.0.5.0/24-32          5 10.0.77.1    valid
    1.0.6.0/24-32          6 10.0.77.1    valid
    1.0.7.0/24-32          7 10.0.77.1    valid
    1.0.8.0/24-32          8 10.0.77.1    valid
    72.9.224.0/19-24       26234 192.168.1.100 valid  *
    72.9.224.0/19-24       3320 192.168.1.200 invalid *
    10.0.0.0/8-32          0 internal    valid

IPv4 records: 14
IPv6 records: 0

```

## show validation group

<b>Syntax</b>	show validation group <instance <i>instance-name</i> > <logical-system <i>logical-system-name</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Display information about route validation redundancy groups.
<b>Options</b>	<p><b>none</b>—Display information about all route validation groups.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information about route validation groups for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show validation group on page 1043</a>
<b>Output Fields</b>	<p><a href="#">Table 40 on page 1042</a> describes the output fields for the <b>show validation group</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 40: show validation group Output Fields**

Field Name	Field Description
Group	Group name.
Maximum sessions	Number of concurrent sessions for each group. The default is 2. The number is configurable with the <a href="#">max-sessions</a> statement.
Session	Resource public key infrastructure (RPKI) cache session IP address.
State	State of the connection between the routing device and the cache server. <b>Up</b> means that the connection is established. <b>Connect</b> means that the connection is not established.
Preference	<p>Each cache server has a preference. Higher preferences are preferred. During a session start or restart, the routing device attempts to start a session with the cache server that has the numerically highest preference. The routing device connects to multiple cache servers in preference order.</p> <p>The default preference is 100. The preference is configurable with the <a href="#">preference</a> statement.</p>

## Sample Output

### show validation group

```
user@host> show validation group
master
  Group: test, Maximum sessions: 3
    Session 10.255.255.11, State: Up, Preference: 100
    Session 10.255.255.12, State: Up, Preference: 100
  Group: test2, Maximum sessions: 2
    Session 10.255.255.13, State: Connect, Preference: 100
```

## show validation replication database

---

<b>Syntax</b>	<pre>show validation replication database &lt;brief   detail&gt; &lt;instance <i>instance-name</i>&gt; &lt;logical-system <i>logical-system-name</i>&gt; &lt;origin-autonomous-system <i>as-number</i>&gt; &lt;record <i>ip-prefix</i>&gt; &lt;session <i>ip-address</i>&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Display the state of the nonstop active routing (NSR) records. The output is the same as the output of the <a href="#">show validation database</a> command, except for the <b>Mismatch</b> column.
<b>Options</b>	<p><b>none</b>—Display all route validation database entries.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information about route validation database entries for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p> <p><b>origin-autonomous-system <i>as-number</i></b>—(Optional) Filter the output by mismatched origin autonomous systems. The <b>mismatch</b> qualifier is useful for finding conflicting origin-autonomous-system information between resource public key infrastructure (RPKI) caches. Mismatches might occur during cache reconfiguration.</p> <p><b>record <i>ip-prefix</i></b>—(Optional) Filter the output by route validation records that match a given prefix.</p> <p><b>session <i>ip-address</i></b>—(Optional) Filter the output by a specific RPKI cache session.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li></ul>
<b>List of Sample Output</b>	<a href="#">show validation replication database on page 1045</a>
<b>Output Fields</b>	<a href="#">Table 41 on page 1045</a> describes the output fields for the <b>show validation replication database</b> command. Output fields are listed in the approximate order in which they appear.

Table 41: show validation replication database Output Fields

Field Name	Field Description	Level of Output
Prefix	Route validation (RV) record prefix.  RV records are received from the cache server and can also be configured statically at the <code>[edit routing-options validation static]</code> hierarchy level.	All levels
Origin-AS	Legitimate originator autonomous system (AS).	All levels
Session	IP address of the RPKI cache server.	All levels
State	State of the route validation records. The state can be <b>valid</b> or <b>invalid</b> .	All levels
IPv4 records	Number of IPv4 route validation records.	All levels
IPv6 records	Number of IPv6 route validation records.	All levels

## Sample Output

### show validation replication database

```

user@host> show validation replication database
RV database for instance master

    Prefix          Origin-AS Session      State
1.0.1.0/24-32      1 10.0.77.1    valid
1.0.2.0/24-32      2 10.0.77.1    valid
1.0.3.0/24-32      3 10.0.77.1    valid
1.0.4.0/24-32      4 10.0.77.1    valid
1.0.5.0/24-32      5 10.0.77.1    valid
1.0.6.0/24-32      6 10.0.77.1    valid
1.0.7.0/24-32      7 10.0.77.1    valid
1.0.8.0/24-32      8 10.0.77.1    valid
72.9.224.0/19-24   26234 192.168.1.100 valid
72.9.224.0/19-24   3320 192.168.1.200 invalid
10.0.0.0/8-32      0 internal    valid

IPv4 records: 14
IPv6 records: 0

```

## show validation session

<b>Syntax</b>	<pre>show validation session &lt;brief   detail&gt; &lt;destination&gt; &lt;instance instance-name&gt; &lt;logical-system logical-system-name&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Display information about all sessions or a specific session with a resource public key infrastructure (RPKI) cache server.
<b>Options</b>	<p><b>none</b>—Display information about all sessions.</p> <p><b>destination</b>—(Optional) Display information about a specific session.</p> <p><b>brief   detail</b>—(Optional) Display the specified level of output.</p> <p><b>instance instance-name</b>—(Optional) Display information about sessions for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system logical-system-name</b>—(Optional) Perform this operation on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show validation session brief on page 1048</a> <a href="#">show validation session detail on page 1048</a>
<b>Output Fields</b>	<p><a href="#">Table 42 on page 1046</a> describes the output fields for the <b>show validation session</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 42: show validation session Output Fields**

Field Name	Field Description	Level of Output
Session	IP address of the RPKI cache server.	All levels
State	State of the connection between the routing device and the cache server. <b>Up</b> means that the connection is established. <b>Connect</b> means that the connection is not established.	All levels
Flaps	Number of attempts to establish a session.	None and <b>brief</b>

Table 42: show validation session Output Fields (*continued*)

Field Name	Field Description	Level of Output
Uptime	Length of time that the session has remained established.	None and brief
#IPv4/IPv6 records	Number of IPv4 and IPv6 route validation records.	None and brief
Session index	Every session has an index number.	detail
Group	Name of the group to which the session belongs	detail
Preference	<p>Each cache server has a preference. Higher preferences are preferred. During a session start or restart, the routing device attempts to start a session with the cache server that has the numerically highest preference. The routing device connects to multiple cache servers in preference order.</p> <p>The default preference is 100. The preference is configurable with the <a href="#">preference</a> statement.</p>	detail
Port	<p>TCP port number for the outgoing connection with the cache server. The well-known RPKI port is TCP port 2222. For a given deployment, an RPKI cache server might listen on some other TCP port number. If so, you can configure the alternative port number with the <a href="#">port</a> statement.</p>	detail
Refresh time	<p>Liveliness check interval for an RPKI cache server. Every <a href="#">refresh-time</a> (seconds), a serial query protocol data unit (PDU) with the last known serial number is transmitted. The <a href="#">hold-time</a> must be at least 2 x the <a href="#">refresh-time</a>.</p>	detail
Hold time	<p>Length of time in seconds that the session between the routing device and the cache server is considered operational without any activity. After the hold time expires, the session is dropped.</p> <p>Reception of any PDU from the cache server resets the hold timer. The <a href="#">hold-time</a> is 600 seconds, by default, and must be least 2 x the <a href="#">refresh-time</a>. If the hold time expires, the session is considered to be down. This, in turn, triggers a session restart event. During a session restart, the routing device attempts to start a session with the cache server that has the numerically highest <a href="#">preference</a>.</p>	detail
Record Life time	<p>Amount of time that route validation (RV) records learned from a cache server are valid. RV records expire if the session to the cache server goes down and remains down for the <a href="#">record-lifetime</a> (seconds).</p>	detail
Serial (Full Update)	Number of full serial updates.	detail
Serial (Incremental Update)	Number of incremental serial updates.	detail
Session flaps	Number of attempts to establish a session.	detail

Table 42: show validation session Output Fields (*continued*)

Field Name	Field Description	Level of Output
Session uptime	Length of time that the session has remained established.	<b>detail</b>
Last PDU received	Time when the most recent PDU was received.	<b>detail</b>
IPv4 prefix count	Number of IPv4 sessions.	<b>detail</b>
IPv6 prefix count	Number of IPv6 sessions.	<b>detail</b>

## Sample Output

### show validation session brief

```

user@host> show validation session brief
Session                               State   Flaps    Uptime #IPv4/IPv6
records
  1.3.0.2                             up      2      00:01:37 13/0
  10.255.255.11                       up      3      00:00:01 1/0
  10.255.255.12                       connect 2      64/68

```

### show validation session detail

```

user@host> show validation session detail
Session 10.0.77.1, State: up
  Group: test, Preference: 100
  Local IPv4 address: 10.0.77.2, Port: 2222
  Refresh time: 300s
  Session flaps: 14, Last Session flap: 5h13m18s ago
  Hold time: 900s
  Record Life time: 3600s
  Serial (Full Update): 0
  Serial (Incremental Update): 0
    Session flaps 2
    Session uptime: 00:48:35
    Last PDU received: 00:03:35
    IPv4 prefix count: 71234
    IPv6 prefix count: 345

```

## show validation statistics

<b>Syntax</b>	show validation statistics <instance <i>instance-name</i> > <logical-system <i>logical-system-name</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 12.2.
<b>Description</b>	Display route validation statistics.
<b>Options</b>	<p><b>none</b>—Display statistics for all routing instances.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display information for the specified routing instance. The instance name can be master for the main instance, or any valid configured instance name or its prefix.</p> <p><b>logical-system <i>logical-system-name</i></b>—(Optional) Perform this operation on a particular logical system.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Origin Validation for BGP on page 473</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show validation statistics on page 1050</a>
<b>Output Fields</b>	<a href="#">Table 43 on page 1049</a> describes the output fields for the <b>show validation statistics</b> command. Output fields are listed in the approximate order in which they appear.

**Table 43: show validation statistics Output Fields**

Field Name	Field Description
Total RV records	Group name.
Total Replication RV records	Number of concurrent sessions for each group. The default is 2. The number is configurable with the <a href="#">max-sessions</a> statement.
Prefix entries	Resource public key infrastructure (RPKI) cache session IP address.
Origin-AS entries	State of the connection between the routing device and the cache server. <b>Up</b> means that the connection is up. <b>Connect</b> means that the connection is not up.
Memory utilization	<p>Each cache server has a preference. Higher preferences are preferred. During a session start or restart, the routing device attempts to start a session with the cache server that has the numerically highest preference. The routing device connects to multiple cache servers in preference order.</p> <p>The default preference is 100. The preference is configurable with the <a href="#">preference</a> statement.</p>

Table 43: show validation statistics Output Fields (*continued*)

Field Name	Field Description
Policy origin-validation requests	Number of queries for validation state of a given instance and prefix.
Valid	Number of valid prefixes reported by the validation query.
Invalid	Number of invalid prefixes reported by the validation query.
Unknown	Number of unknown prefixes reported by the validation query. This means that the prefix is not found in the database.
BGP import policy reevaluation notifications	A change, addition, or deletion of a route validation record triggers a BGP import reevaluation for all exact matching and more specific prefixes.
inet.0	Number of IPv4 route validation records that have been added, deleted, or changed.
inet6.0	Number of IPv6 route validation records that have been added, deleted, or changed.

## Sample Output


### show validation statistics

```

user@host> show validation statistics
Total RV records:          453455
Total Replication RV records: 453455
  Prefix entries:          35432
  Origin-AS entries:       124400
Memory utilization: 16.31MB
Policy origin-validation requests: 234995
  valid:                    23445
  invalid:                  14666
  unknown:                  34567
BGP import policy reevaluation notifications: 460268
  inet.0:                   435345
  inet6.0:                   3454

```

## test policy

<b>Syntax</b>	<code>test policy <i>policy-name</i> <i>prefix</i></code>
<b>Release Information</b>	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches.
<b>Description</b>	Test a policy configuration to determine which prefixes match routes in the routing table.
<div>  <p><b>NOTE:</b> If you are using the <code>test policy</code> command on a logical system, you must first set the CLI to the logical system context. For example, if you want to test a routing policy that is configured on logical system R2, first run the <code>set cli logical-system R2</code> command.</p> </div>	
<b>Options</b>	<p><i>policy-name</i>—Name of a policy.</p> <p><i>prefix</i>—Destination prefix to match.</p>
<b>Additional Information</b>	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 <b>from</b> clause in the specified policy. All prefixes accepted by the policy are displayed. The <b>test policy</b> command evaluates a policy differently from the BGP import process. When testing a policy that contains an <b>interface</b> match condition in the <b>from</b> clause, the <b>test policy</b> command uses the match condition. In contrast, BGP does not use the <b>interface</b> match condition when evaluating the policy against routes learned from internal BGP (IBGP) or external BGP (EGBP) multihop peers.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding Routing Policy Tests</i></li> <li>• <i>Example: Testing a Routing Policy with Complex Regular Expressions</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">test policy on page 1051</a>
<b>Output Fields</b>	For information about output fields, see the output field tables for the <a href="#">show route</a> command, the <a href="#">show route detail</a> command, the <a href="#">show route extensive</a> command, or the <a href="#">show route terse</a> 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 12
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

- [BGP Troubleshooting on page 1055](#)
- [Routing Protocol Process Memory FAQs on page 1107](#)



## CHAPTER 17

# BGP Troubleshooting

- [Understanding Hidden Routes on page 1056](#)
- [Checklist for Verifying the BGP Protocol and Peers on page 1057](#)
- [Checklist for Checking the BGP Layer on page 1058](#)
- [Checking the BGP Layer on page 1058](#)
- [Check BGP Sessions on page 1072](#)
- [Verify BGP Peers on page 1074](#)
- [Verify the BGP Protocol on page 1075](#)
- [Verify the BGP Configuration on page 1077](#)
- [Display Sent or Received BGP Packets on page 1082](#)
- [Diagnose BGP Session Establishment Problems on page 1083](#)
- [Examine BGP Routes and Route Selection on page 1084](#)
- [Examine Routes in the Forwarding Table on page 1090](#)
- [Log BGP State Transition Events on page 1091](#)
- [Display Detailed BGP Protocol Information on page 1093](#)
- [Verify Received BGP Routes on page 1095](#)
- [Verify That a Particular BGP Route Is Received on Your Router on page 1096](#)
- [Verifying Advertised BGP Routes on page 1096](#)
- [Check That BGP Traffic Is Using the LSP on page 1097](#)
- [Check That BGP Traffic Is Using the LSP Again on page 1097](#)
- [Examine the EBGp over IBGP Selection on page 1098](#)
- [Verify BGP on an Internal Router on page 1099](#)
- [Verify BGP on a Border Router on page 1102](#)

## Understanding Hidden Routes

---

Hidden routes are routes that the device cannot use for reasons such as an invalid next hop or a routing policy that rejects the routes.



**NOTE:** If a route is completely invalid, the route is not placed into the routing table as a candidate route and does not even appear as hidden.

Following are some useful commands for viewing and troubleshooting hidden routes:

- `show route hidden (terse | detail | extensive)`
- `show route hidden-route extensive`
- `show route next-hop-of-hidden-route extensive`
- `show route resolution unresolved detail`

Routes can be hidden for the following reasons:

- An import policy rejects the route.
- The next hop cannot be resolved using the current indirect next hop resolution rule. Because routing protocols such as internal BGP (IBGP) can send routing information about indirectly connected routes, Junos OS relies on routes from intra-AS routing protocols (OSPF, IS-IS, RIP, and static) to resolve the best directly connected next hop. The Routing Engine performs route resolution to determine the best directly connected next hop and installs the route to the Packet Forwarding Engine.
- A damping policy suppresses the route.
- The route reflector cluster ID is looped. If a BGP router that receives a route from an IBGP neighbor is configured to operate as a route reflector and in the incoming update detects the presence of its own cluster ID in the cluster-list attribute, it will reject the update.
- The confederation sub-AS number is looped.
- The AS path contains illegal or invalid confederation attributes.
- An AS number is looped in the AS path.
- The originator ID is looped. If a BGP router that receives a route from an IBGP neighbor in the incoming update detects the presence of its own router ID in the originator ID attribute, it will reject the update.
- The next hop address is the address of the local routing device.
- The AS path contains illegal or invalid transitive attributes.
- The AS path is empty. This only applies to EBGP. For IBGP, an empty AS path is normal.
- The AS path contains a zero.
- The next hop address is a multicast address.

- The next hop address is an IPv6 link-local address.
- The route prefix or the route next hop is a martian address.

**Related Documentation**

- *Examples: Configuring Static Routes*
- *Example: Enabling Indirect Next Hops on the Packet Forwarding Engine*
- [Example: Configuring BGP Route Reflectors on page 419](#)
- [Example: Configuring BGP Confederations on page 441](#)
- [Examples: Configuring BGP Flap Damping on page 505](#)

## Checklist for Verifying the BGP Protocol and Peers

**Purpose** Table 44 on page 1057 provides links and commands for verifying whether the Border Gateway Protocol (BGP) is configured correctly on a Juniper Networks router in your network, the internal Border Gateway Protocol (IBGP) and exterior Border Gateway Protocol (EBGP) sessions are properly established, the external routes are advertised and received correctly, and the BGP path selection process is working properly.

**Action**

**Table 44: Checklist for Verifying the BGP Protocol and Peers**

Tasks	Command or Action
<b>"Verify the BGP Protocol" on page 1075</b>	
1. <a href="#">Verify BGP on an Internal Router on page 1099</a>	<code>show configuration</code>
2. <a href="#">Verify BGP on a Border Router on page 1102</a>	<code>show configuration</code>
<b>"Verify BGP Peers" on page 1074</b>	
1. <a href="#">Check BGP Sessions on page 1061</a>	<code>show bgp summary</code>
2. <a href="#">Verifying Advertised BGP Routes on page 1096</a>	<code>show route advertising-protocol bgp <i>neighbor-address</i></code>
3. <a href="#">Verify That a Particular BGP Route Is Received on Your Router on page 1096</a>	<code>show route receive-protocol bgp <i>neighbor-address</i></code>
<b>"Examine BGP Routes and Route Selection" on page 1084</b>	
1. <a href="#">Examine the Local Preference Selection on page 1086</a>	<code>show route <i>destination-prefix</i> &lt; detail &gt;</code>
2. <a href="#">Examine the Multiple Exit Discriminator Route Selection on page 1087</a>	<code>show route <i>destination-prefix</i> &lt; detail &gt;</code>
3. <a href="#">Examine the EBGp over IBGP Selection on page 1088</a>	<code>show route <i>destination-prefix</i> &lt; detail &gt;</code>
4. <a href="#">Examine the IGP Cost Selection on page 1089</a>	<code>show route <i>destination-prefix</i> &lt; detail &gt;</code>
<b>"Examine Routes in the Forwarding Table" on page 1090</b>	<code>show route forwarding-table</code>

## Checklist for Checking the BGP Layer

**Problem** **Description:** This checklist provides the steps and commands for checking the BGP configuration of the Multiprotocol Label Switching (MPLS) network. The checklist provides links to an overview of the BGP configuration and more detailed information about the commands used to configure BGP. (See [Table 45 on page 1058](#).)

**Table 45: Checklist for Checking the BGP Layer**

Tasks	Command or Action
<b>“Checking the BGP Layer” on page 1058</b>	
1. <a href="#">Check That BGP Traffic Is Using the LSP on page 1060</a>	<code>traceroute <i>hostname</i></code>
2. <a href="#">Check BGP Sessions on page 1061</a>	<code>show bgp summary</code>
3. <a href="#">Verify the BGP Configuration on page 1062</a>	<code>show configuration</code>
4. <a href="#">Examine BGP Routes on page 1068</a>	<code>show route <i>destination-prefix</i> detail</code>
5. <a href="#">Verify Received BGP Routes on page 1069</a>	<code>show route receive protocol bgp <i>neighbor-address</i></code>
6. <a href="#">Take Appropriate Action on page 1070</a>	<p>The following sequence of commands addresses the specific problem described in this topic:</p> <pre> [edit] edit protocols bgp  [edit protocols bgp] show set local-address 10.0.0.1 delete group internal neighbor 10.1.36.2 show commit </pre>
7. <a href="#">Check That BGP Traffic Is Using the LSP Again on page 1071</a>	<code>traceroute <i>hostname</i></code>

## Checking the BGP Layer

**Purpose** After you have configured the label-switched path (LSP) and determined that it is up, and configured BGP and determined that sessions are established, ensure that BGP is using the LSP to forward traffic.

[Figure 69 on page 1059](#) illustrates the BGP layer of the layered MPLS model.

Figure 69: Checking the BGP Layer

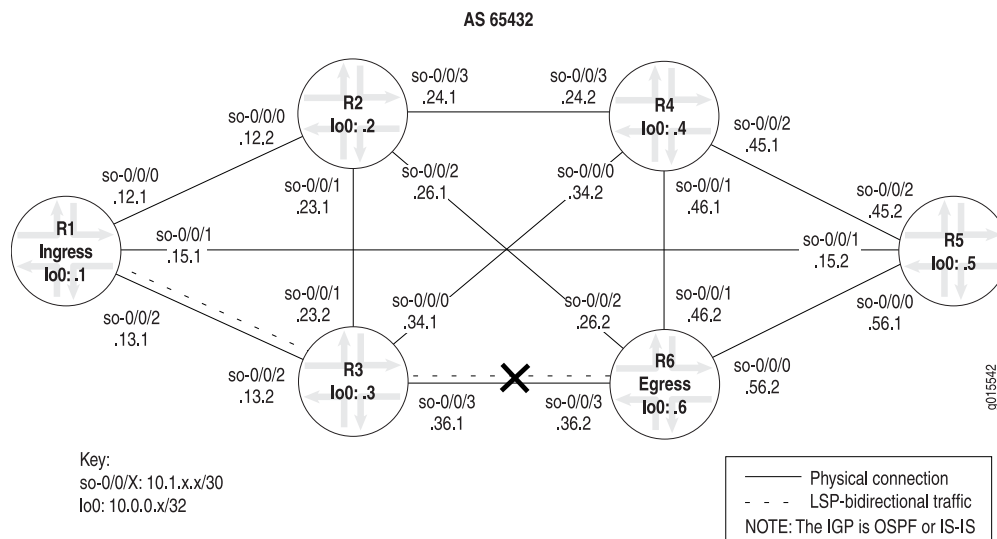
<b>BGP Layer</b>	<traceroute <i="">host-name  show bgp summary  show configuration protocols bgp  show route <i>destination-prefix</i> detail  show route receive protocol bgp <i>neighbor-address</i> </traceroute>
<b>MPLS Layer</b>	show mpls lsp show mpls lsp extensive show route table mpls.0 show route <i>address</i> traceroute <i>address</i> ping mpls rsvp <i>lsp-name</i> detail
<b>RSVP Layer</b>	show rsvp session show rsvp neighbor show rsvp interface
<div>↙ IGP and IP Layers Functioning ↘</div>	
<b>OSPF Layer</b>	<b>IS-IS Layer</b>
show ospf neighbor show configuration protocols ospf show ospf interface	show isis adjacency show configuration protocols isis show isis interface
<b>IP Layer</b>	<b>IP Layer</b>
show ospf neighbor extensive show interfaces terse	show isis adjacency extensive show interfaces terse
<b>Data Link Layer</b>	show interfaces extensive <i>"JUNOS Interfaces Operations Guide"</i>
<b>Physical Layer</b>	show interfaces show interfaces terse ping <i>host</i>

9015548

When you check the BGP layer, you verify that the route is present and active, and more importantly, you ensure that the next hop is the LSP. There is no point in checking the BGP layer unless the LSP is established, because BGP uses the MPLS LSP to forward traffic. If the network is not functioning at the BGP layer, the LSP does not work as configured.

Figure 70 on page 1060 illustrates the MPLS network used in this topic.

Figure 70: MPLS Network Broken at the BGP Layer



The network shown in [Figure 70 on page 1060](#) is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6** through **R3** to **R1**, creating bidirectional traffic.

The cross shown in [Figure 70 on page 1060](#) indicates where BGP is not being used to forward traffic through the LSP. Possible reasons for the LSP not working correctly are that the destination IP address of the LSP does not equal the BGP next hop or that BGP is not configured properly.

To check the BGP layer, follow these steps:

1. [Check That BGP Traffic Is Using the LSP on page 1060](#)
2. [Check BGP Sessions on page 1061](#)
3. [Verify the BGP Configuration on page 1062](#)
4. [Examine BGP Routes on page 1068](#)
5. [Verify Received BGP Routes on page 1069](#)
6. [Take Appropriate Action on page 1070](#)
7. [Check That BGP Traffic Is Using the LSP Again on page 1071](#)

## Check That BGP Traffic Is Using the LSP

**Purpose** At this level of the troubleshooting model, BGP and the LSP may be up, however BGP traffic might not be using the LSP to forward traffic.

**Action** To verify that BGP traffic is using the LSP, enter the following Junos OS command-line interface (CLI) operational mode command from the ingress router:

```
user@host> traceroute hostname
```

## Sample Output

```

user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1 10.1.13.2 (10.1.13.2) 0.653 ms 0.590 ms 0.543 ms
 2 10.1.36.2 (10.1.36.2) 0.553 ms !N 0.552 ms !N 0.537 ms !N

user@R6> traceroute 100.100.1.1
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
 1 10.1.36.1 (10.1.36.1) 0.660 ms 0.551 ms 0.526 ms
 2 10.1.13.1 (10.1.13.1) 0.568 ms !N 0.553 ms !N 0.536 ms !N

```

**Meaning** The sample output shows that BGP traffic is not using the LSP, consequently MPLS labels do not appear in the output. Instead of using the LSP, BGP traffic is using the interior gateway protocol (IGP) to reach the BGP next-hop LSP egress address for **R6** and **R1**. The Junos OS default is to use LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

## Check BGP Sessions

**Purpose** Display summary information about BGP and its neighbors to determine if routes are received from peers in the autonomous system (AS). When a BGP session is established, the peers are exchanging update messages.

**Action** To check that BGP sessions are up, enter the following Junos OS CLI operational mode command from the ingress router:

```
user@host> show bgp summary
```

## Sample Output 1

```

user@R1> show bgp summary
Groups: 1 Peers: 6 Down peers: 1
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 1 1 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2 65432 11257 11259 0 0 3d 21:49:57 0/0/0
0/0/0
10.0.0.3 65432 11257 11259 0 0 3d 21:49:57 0/0/0
0/0/0
10.0.0.4 65432 11257 11259 0 0 3d 21:49:57 0/0/0
0/0/0
10.0.0.5 65432 11257 11260 0 0 3d 21:49:57 0/0/0
0/0/0
10.0.0.6 65432 4 4572 0 13d 21:46:59 Active
10.1.36.2 65432 11252 11257 0 0 3d 21:46:49 1/1/0
0/0/0

```

## Sample Output 2

```

user@R1> show bgp summary
Groups: 1 Peers: 5 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 1 1 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn

```

	State #Active/Received/Damped...
10.0.0.2	65432 64 68 0 0 32:18 0/0/0
10.0.0.3	65432 64 67 0 0 32:02 0/0/0
10.0.0.4	65432 64 67 0 0 32:10 0/0/0
10.0.0.5	65432 64 67 0 0 32:14 0/0/0
10.0.0.6	65432 38 39 0 1 18:02 1/1/0

**Meaning** Sample Output 1 shows that one peer (egress router **10.0.0.6**) is not established, as indicated by the **Down Peers: 1** field. The last column (**State|#Active/Received/Damped**) shows that peer **10.0.0.6** is active, indicating that it is not established. All other peers are established as indicated by the number of active, received, and damped routes. For example, **0/0/0** for peer **10.0.0.2** indicates that no BGP routes were active or received in the routing table, and no BGP routes were damped; **1/1/0** for peer **10.1.36.2** indicates that one BGP route was active and received in the routing table, and no BGP routes were damped.

If the output of the **show bgp summary** command of an ingress router shows that a neighbor is down, check the BGP configuration. For information on checking the BGP configuration, see [“Verify the BGP Configuration” on page 1062](#).

Sample Output 2 shows output from ingress router **R1** after the BGP configurations on **R1** and **R6** were corrected in [“Take Appropriate Action” on page 1070](#). All BGP peers are established and one route is active and received. No BGP routes were damped.

If the output of the **show bgp summary** command shows that a neighbor is up but packets are not being forwarded, check for received routes from the egress router. For information on checking the egress router for received routes, see [“Verify Received BGP Routes” on page 1069](#).

## Verify the BGP Configuration

**Purpose** For BGP to run on the router, you must define the local AS number, configure at least one group, and include information about at least one peer in the group (the peer's IP address and AS number). When BGP is part of an MPLS network, you must ensure that the LSP is configured with a destination IP address equal to the BGP next hop in order for BGP routes to be installed with the LSP as the next hop for those routes.

**Action** To verify the BGP configuration, enter the following Junos OS CLI operational mode command:

```
user@host> show configuration
```

## Sample Output 1

```
user@R1> show configuration
[...Output truncated...]
interfaces {
  so-0/0/0 {
    unit 0 {
```

```

        family inet {
            address 10.1.12.1/30;
        }
        family iso;
        family mpls;
    }
}
so-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.15.1/30;
        }
        family iso;
        family mpls;
    }
}
so-0/0/2 {
    unit 0 {
        family inet {
            address 10.1.13.1/30;
        }
        family iso;
        family mpls;
    }
}
fxp0 {
    unit 0 {
        family inet {
            address 192.168.70.143/21;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.1/32;
        }
        family iso {
            address 49.0004.1000.0000.0001.00;
        }
    }
}
}
routing-options {
    [...Output truncated...]
    route 100.100.1.0/24 reject;
}
router-id 10.0.0.1;
autonomous-system 65432;
}
protocols {
    rsvp {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface fxp0.0 {
            disable;
        }
    }
}
mpls {
    label-switched-path R1-to-R6 {

```

```

        to 10.0.0.6; <<< destination address of the LSP
    }
    inactive: interface so-0/0/0.0;
    inactive: interface so-0/0/1.0;
    interface so-0/0/2.0;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    export send-statics; <<< missing local-address statement
    group internal {
        type internal;
        neighbor 10.0.0.2;
        neighbor 10.0.0.5;
        neighbor 10.0.0.4;
        neighbor 10.0.0.6;
        neighbor 10.0.0.3;
        neighbor 10.1.36.2; <<< incorrect interface address
    }
}
isis {
    level 1 disable;
    interface so-0/0/0.0;
    interface so-0/0/1.0;
    interface so-0/0/2.0;
    interface all {
        level 2 metric 10;
    }
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
        passive;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface lo0.0; {
            passive
        }
    }
}
}
policy-options {
    policy-statement send-statics {
        term statics {
            from {
                route-filter 100.100.1.0/24 exact;
            }
            then accept;
        }
    }
}
}

```

## Sample Output 2

```

user@R6> show configuration
[...Output truncated...]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.46.2/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/2 {
    unit 0 {
      family inet {
        address 10.1.26.2/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.1.36.2/30;
      }
      family iso;
      family mpls;
    }
  }
  fxp0 {
    unit 0 {
      family inet {
        address 192.168.70.148/21;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.6/32;
        address 127.0.0.1/32;
      }
      family iso {
        address 49.0004.1000.0000.0006.00;
      }
    }
  }
}

```

```

routing-options {
  [...Output truncated...]
  route 100.100.6.0/24 reject;
}
router-id 10.0.0.6;
autonomous-system 65432;
}
protocols {
  rsvp {
    interface so-0/0/0.0;
    interface so-0/0/1.0;
    interface so-0/0/2.0;
    interface so-0/0/3.0;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path R6-to-R1 {
      to 10.0.0.1; <<< destination address of the reverse LSP
    }
    inactive: interface so-0/0/0.0;
    inactive: interface so-0/0/1.0;
    inactive: interface so-0/0/2.0;
    interface so-0/0/3.0;
  }
  bgp {
    group internal {
      type internal;
      export send-statics; <<< missing local-address statement
      neighbor 10.0.0.2;
      neighbor 10.0.0.3;
      neighbor 10.0.0.4;
      neighbor 10.0.0.5;
      neighbor 10.0.0.1;
      neighbor 10.1.13.1; <<< incorrect interface address
    }
  }
  isis {
    level 1 disable;
    interface all {
      level 2 metric 10;
    }
    interface fxp0.0 {
      disable;
    }
    interface lo0.0 {
      passive;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface so-0/0/1.0;
      interface so-0/0/2.0;
      interface so-0/0/3.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}

```

```

    }
  }
  policy-options {
    policy-statement send-statics {
      term statics {
        from {
          route-filter 100.100.6.0/24 exact;
        }
        then accept;
      }
    }
  }
}

```

**Meaning** The sample output shows the BGP configurations on ingress router **R1** and egress router **R6**. Both configurations show the local AS (**65432**), one group (**internal**), and six peers configured. The underlying interior gateway protocol is IS-IS, and the relevant interfaces are configured to run IS-IS.



**NOTE:** In this configuration, the RID is manually configured to avoid any duplicate RID problems, and all interfaces configured with BGP include the **family inet** statement at the [edit interfaces *type-fpc/pic/port* unit *logical-unit-number*] hierarchy level.

Sample output for ingress router **R1** and egress router **R6** shows that the BGP protocol configuration is missing the **local-address** statement for the internal group. When the **local-address** statement is configured, BGP packets are forwarded from the local router loopback (**lo0**) interface address, which is the address to which BGP peers are peering. If the **local-address** statement is not configured, BGP packets are forwarded from the outgoing interface address, which does not match the address to which BGP peers are peering, and BGP does not come up.

On the ingress router, the IP address (**10.0.0.1**) in the **local-address** statement should be the same as the address configured for the LSP on the egress router (**R6**) in the **to** statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level. BGP uses this address, which is identical to the LSP address, to forward BGP traffic through the LSP.

In addition, the BGP configuration on **R1** includes two IP addresses for **R6**, an interface address (**10.1.36.2**) and a loopback (**lo0**) interface address (**10.0.0.6**), resulting in the LSP destination address (**10.0.0.6**) not matching the BGP next-hop address (**10.1.36.2**). The BGP configuration on **R6** also includes two IP addresses for **R1**, an interface address (**10.1.13.1**) and a loopback (**lo0**) interface address, resulting in the reverse LSP destination address (**10.0.0.1**) not matching the BGP next-hop address (**10.1.13.1**).

In this instance, because the **local-address** statement is missing in the BGP configurations of both routers and the LSP destination address does not match the BGP next-hop address, BGP is not using the LSP to forward traffic.

## Examine BGP Routes

**Purpose** You can examine the BGP path selection process to determine the single, active path when BGP receives multiple routes to the same destination. In this step, we examine the reverse LSP **R6-to-R1**, making **R6** the ingress router for that LSP.

**Action** To examine BGP routes and route selection, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix detail
```

### Sample Output 1

```
user@R6> show route 100.100.1.1 detail
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
100.100.1.0/24 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
             Source: 10.1.13.1
             Next hop: via so-0/0/3.0, selected
               Protocol next hop: 10.1.13.1 Indirect next hop: 8671594 304
             State: <Active Int Ext>
             Local AS: 65432 Peer AS: 65432
             Age: 4d 5:15:39 Metric2: 2
             Task: BGP_65432.10.1.13.1+3048
             Announcement bits (2): 0-KRT 4-Resolve inet.0
             AS path: I
             Localpref: 100
             Router ID: 10.0.0.1
```

### Sample Output 2

```
user@R6> show route 100.100.1.1 detail
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
100.100.1.0/24 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
             Source: 10.0.0.1
             Next hop: via so-0/0/3.0 weight 1, selected
               Label-switched-path R6-to-R1
             Label operation: Push 100000
               Protocol next hop: 10.0.0.1 Indirect next hop: 8671330 301
             State: <Active Int Ext>
             Local AS: 65432 Peer AS: 65432
             Age: 24:35 Metric2: 2
             Task: BGP_65432.10.0.0.1+179
             Announcement bits (2): 0-KRT 4-Resolve inet.0
             AS path: I
             Localpref: 100
             Router ID: 10.0.0.1
```

**Meaning** Sample Output 1 shows that the BGP next hop (10.1.13.1) does not equal the LSP destination address (10.0.0.1) in the **to** statement at the **[edit protocols mpls label-switched-path label-switched-path-name]** hierarchy level when the BGP configuration of **R6** and **R1** is incorrect.

Sample Output 2, taken after the configurations on R1 and R6 are corrected, shows that the BGP next hop (10.0.0.1) and the LSP destination address (10.0.0.1) are the same, indicating that BGP can use the LSP to forward BGP traffic.

## Verify Received BGP Routes

**Purpose** Display the routing information received on router **R6**, the ingress router for the reverse LSP **R6-to-R1**.

**Action** To verify that a particular BGP route is received on the egress router, enter the following Junos OS CLI operational mode command:

```
user@host> show route receive protocol bgp neighbor-address
```

## Sample Output 1

```
user@R6> show route receive-protocol bgp 10.0.0.1
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
<<< missing route
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: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0
hidden)
```

## Sample Output 2

```
user@R6> show route receive-protocol bgp 10.0.0.1
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
  Prefix                Nexthop          MED      Lc1pref  AS path
*100.100.1.0/24    10.0.0.1          100      I

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: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0
hidden)
```

**Meaning** Sample Output 1 shows that ingress router **R6** (reverse LSP **R6-to-R1**) does not receive any BGP routes into the **inet.0** routing table when the BGP configurations of **R1** and **R6** are incorrect.

Sample Output 2 shows a BGP route installed in the **inet.0** routing table after the BGP configurations on **R1** and **R6** are corrected using [“Take Appropriate Action” on page 1070](#).

## Take Appropriate Action

**Problem**    **Description:** Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In this example, the ingress and egress routers are incorrectly configured for BGP to forward traffic using the LSP.

**Solution**    To correct the errors in this example, follow these steps:

1. On ingress router R1, include the **local-address** statement and delete the incorrect interface address (repeat these steps on egress router R6):

```
[edit]
user@R1# edit protocols bgp
[edit protocols bgp]
user@R1# show
user@R1# set local-address 10.0.0.1
user@R1# delete group internal neighbor 10.1.36.2
```

2. Verify and commit the configuration:

```
[edit protocols bgp]
user@R1# show
user@R1# commit
```

**Sample Output**

```
[edit]
user@R1# edit protocols bgp

[edit protocols bgp]
user@R1# show
export send-statics;
group internal {
    type internal;
    neighbor 10.0.0.2;
    neighbor 10.0.0.5;
    neighbor 10.0.0.4;
    neighbor 10.0.0.6;
    neighbor 10.0.0.3;
    neighbor 10.1.36.2;
}

[edit protocols bgp]
user@R1# set local-address 10.0.0.1

[edit protocols bgp]
user@R1# delete group internal neighbor 10.1.36.2

[edit protocols bgp]
user@R1# show
local-address 10.0.0.1;
export send-statics;
group internal {
    type internal;
    neighbor 10.0.0.2;
    neighbor 10.0.0.5;
    neighbor 10.0.0.4;
    neighbor 10.0.0.6;
    neighbor 10.0.0.3;
}

[edit protocols bgp]
user@R1# commit
commit complete
```

**Meaning** The sample output shows that the configuration of BGP on ingress router **R1** is now correct. BGP can now forward BGP traffic through the LSP.

## Check That BGP Traffic Is Using the LSP Again

**Purpose** After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that BGP traffic is using the LSP and that the problem in the BGP layer has been resolved.

**Action** To verify that BGP traffic is using the LSP, enter the following Junos OS CLI operational mode command from the ingress router:

```
user@host> traceroute hostname
```

## Sample Output

```
user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
```

```

1 10.1.13.2 (10.1.13.2) 0.858 ms 0.740 ms 0.714 ms
   MPLS Label=100016 CoS=0 TTL=1 S=1
2 10.1.36.2 (10.1.36.2) 0.592 ms !N 0.564 ms !N 0.548 ms !N

```

```
user@R6> traceroute 100.100.1.1
```

```
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
```

```

1 10.1.36.1 (10.1.36.1) 0.817 ms 0.697 ms 0.771 ms
   MPLS Label=100000 CoS=0 TTL=1 S=1
2 10.1.13.1 (10.1.13.1) 0.581 ms !N 0.567 ms !N 0.544 ms !N

```

**Meaning** The sample output shows that MPLS labels are used to forward packets through the LSP. Included in the output is a label value (**MPLS Label=100016**), the time-to-live value (**TTL=1**), and the stack bit value (**S=1**).

The **MPLS Label** field is used to identify the packet to a particular LSP. It is a 20-bit field, with a maximum value of ( $2^{20}-1$ ), approximately 1,000,000.

The time-to-live (TTL) value contains a limit on the number of hops that this MPLS packet can travel through the network (1). It is decremented at each hop, and if the TTL value drops below one, the packet is discarded.

The bottom of the stack bit value (**S=1**) indicates that is the last label in the stack and that this MPLS packet has one label associated with it. The MPLS implementation in the Junos OS supports a stacking depth of 3 on the M-series routers and up to 5 on the T-series routing platforms. For more information on MPLS label stacking, see RFC 3032, *MPLS Label Stack Encoding*.

MPLS labels appear in the sample output because the **traceroute** command is issued to a BGP destination where the BGP next hop for that route is the LSP egress address. The Junos OS by default uses LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

If the BGP next hop does not equal the LSP egress address, the BGP traffic does not use the LSP, and consequently MPLS labels do not appear in the output for the **traceroute** command, as indicated in the sample output in [“Check BGP Sessions” on page 1061](#).

## Check BGP Sessions

**Purpose** Display summary information about BGP and its neighbors to determine if routes are received from peers in the autonomous system (AS). When a BGP session is established, the peers are exchanging update messages.

**Action** To check that BGP sessions are up, enter the following Junos OS CLI operational mode command from the ingress router:

```
user@host> show bgp summary
```

## Sample Output 1

```
user@R1> show bgp summary
```

```
Groups: 1 Peers: 6 Down peers: 1
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	1	1	0	0	0	0	0
Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	

```

State|#Active/Received/Damped...
10.0.0.2      65432      11257      11259      0      0 3d 21:49:57 0/0/0
              0/0/0
10.0.0.3      65432      11257      11259      0      0 3d 21:49:57 0/0/0
              0/0/0
10.0.0.4      65432      11257      11259      0      0 3d 21:49:57 0/0/0
              0/0/0
10.0.0.5      65432      11257      11260      0      0 3d 21:49:57 0/0/0
              0/0/0
10.0.0.6 65432 4 4572 0 13d 21:46:59 Active
10.1.36.2     65432      11252      11257      0      0 3d 21:46:49 1/1/0
              0/0/0

```

## Sample Output 2

```

user@R1> show bgp summary
Groups: 1 Peers: 5 Down peers: 0
Table      Tot Paths  Act Paths  Suppressed  History  Damp  State  Pending
inet.0      1          1          0           0        0     0       0
Peer      AS      InPkt    OutPkt    OutQ    Flaps  Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2   65432      64        68        0        0      32:18 0/0/0
              0/0/0
10.0.0.3   65432      64        67        0        0      32:02 0/0/0
              0/0/0
10.0.0.4   65432      64        67        0        0      32:10 0/0/0
              0/0/0
10.0.0.5   65432      64        67        0        0      32:14 0/0/0
              0/0/0
10.0.0.6   65432      38        39        0        1      18:02 1/1/0
              0/0/0

```

**Meaning** Sample Output 1 shows that one peer (egress router **10.0.0.6**) is not established, as indicated by the **Down Peers: 1** field. The last column (**State|#Active/Received/Damped**) shows that peer **10.0.0.6** is active, indicating that it is not established. All other peers are established as indicated by the number of active, received, and damped routes. For example, **0/0/0** for peer **10.0.0.2** indicates that no BGP routes were active or received in the routing table, and no BGP routes were damped; **1/1/0** for peer **10.1.36.2** indicates that one BGP route was active and received in the routing table, and no BGP routes were damped.

If the output of the **show bgp summary** command of an ingress router shows that a neighbor is down, check the BGP configuration. For information on checking the BGP configuration, see [“Verify the BGP Configuration” on page 1062](#).

Sample Output 2 shows output from ingress router **R1** after the BGP configurations on **R1** and **R6** were corrected in [“Take Appropriate Action” on page 1070](#). All BGP peers are established and one route is active and received. No BGP routes were damped.

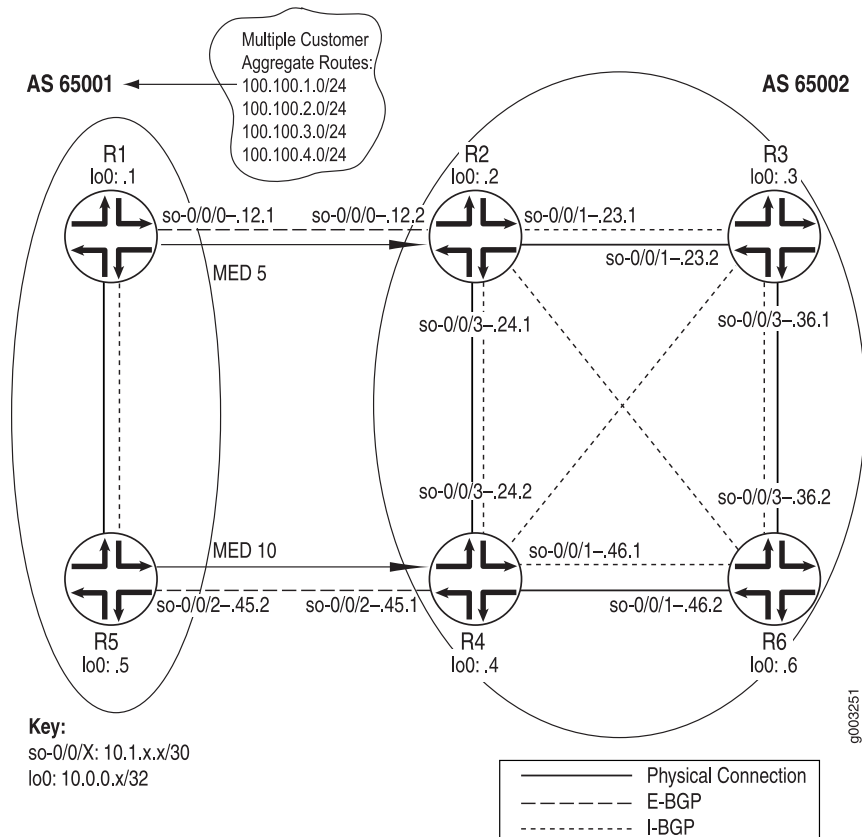
If the output of the **show bgp summary** command shows that a neighbor is up but packets are not being forwarded, check for received routes from the egress router. For information on checking the egress router for received routes, see [“Verify Received BGP Routes” on page 1069](#).

## Verify BGP Peers

**Purpose** Assuming that all the routers are correctly configured for BGP, you can verify if IBGP and EBGP sessions are properly established, external routes are advertised and received correctly, and the BGP path selection process is working properly.

Figure 71 on page 1074 illustrates an example BGP network topology used in this topic.

Figure 71: BGP Network Topology



The network consists of two directly connected ASes consisting of external and internal peers. The external peers are directly connected through a shared interface and are running EBGP. The internal peers are connected through their loopback (lo0) interfaces through IBGP. AS 65001 is running OSPF and AS 65002 is running IS-IS as its underlying IGP. IBGP routers do not have to be directly connected, the underlying IGP allows neighbors to reach one another.

The two routers in AS 65001 each contain one EBGP link to AS 65002 (R2 and R4) over which they announce aggregated prefixes: 100.100.1.0, 100.100.2.0, 100.100.3.0, and 100.100.4.0. Also, R1 and R5 are injecting multiple exit discriminator (MED) values of 5 and 10, respectively, for some routes.

The internal routers in both ASes are using a full mesh IBGP topology. A full mesh is required because the networks are not using confederations or route reflectors, so any routes

learned through IBGP are not distributed to other internal neighbors. For example, when **R3** learns a route from **R2**, **R3** does not distribute that route to **R6** because the route is learned through IBGP, so **R6** must have a direct BGP connection to **R2** to learn the route.

In a full mesh topology, only the border router receiving external BGP information distributes that information to other routers within its AS. The receiving router does not redistribute that information to other IBGP routers in its own AS.

From the point of view of AS 65002, the following sessions should be up:

- The four routers in AS 65002 should have IBGP sessions established with each other.
- **R2** should have an EBGP session established with **R1**.
- **R4** should have an EBGP session established with **R5**.

To verify BGP peers, follow these steps:

---

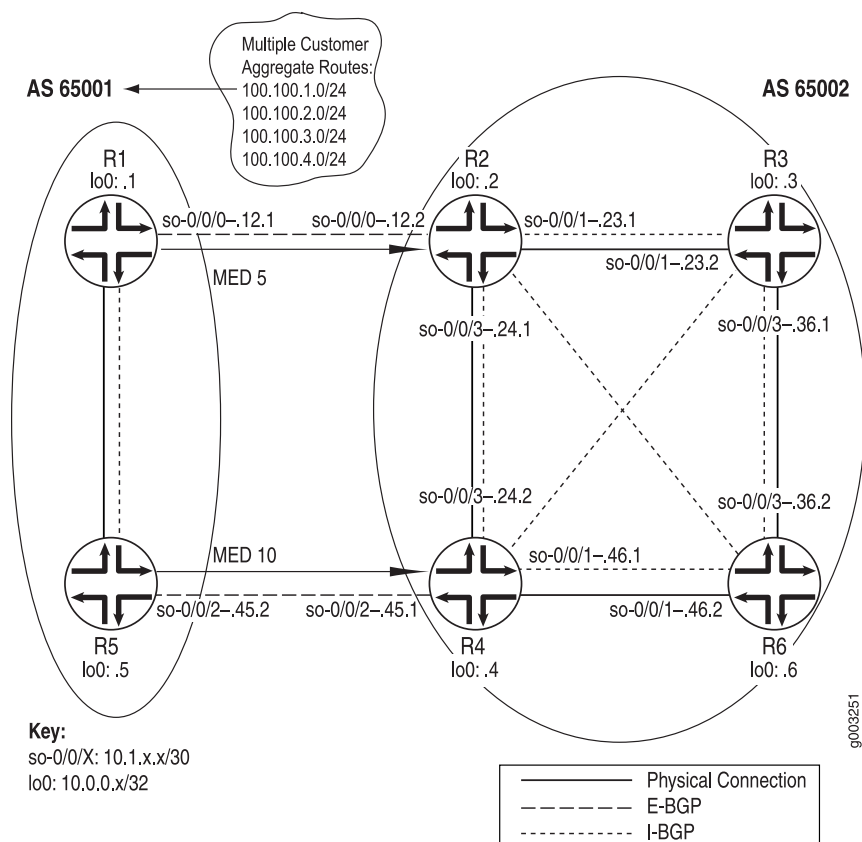
## Verify the BGP Protocol

---

**Purpose** For BGP to run on a router in your network, you must define the local autonomous system (AS) number, configure at least one group, and include information about at least one peer in the group. If the peer is an EBGP peer, include the peer's AS number. For all peers, include either the peer's interface IP address or loopback (**lo0**) IP address. When configuring BGP on an interface, you must also include the **family inet** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

[Figure 72 on page 1076](#) illustrates the example configurations used in this topic.

Figure 72: BGP Configuration Topology



The network in [Figure 72 on page 1076](#) consists of two directly connected ASs. IP addresses included in the network diagram are as follows:

- 10.1.12.1—AS 650001 external IP address on R1
- 10.1.45.2—AS 650001 external IP address on R5
- 10.0.0.1—Internal loopback (lo0) IP address for R1
- 10.0.0.5—Internal loopback (lo0) IP address for R5
- 10.1.12.2—AS 65002 external IP address on R2
- 10.1.45.1—AS 65002 external IP address on R5
- 10.0.0.2—Internal loopback (lo0) address for R2
- 10.0.0.3—Internal loopback (lo0) address for R3
- 10.0.0.4—Internal loopback (lo0) address for R4
- 10.0.0.6—Internal loopback (lo0) address for R6

All routers within each AS maintain an IBGP session between each router in that AS. R1 and R5 have an IBGP session through their loopback (lo0) interfaces: 10.0.0.1 and 10.0.0.5. R2, R3, R4, and R6 maintain IBGP sessions between each other through their loopback (lo0) interfaces: 10.0.0.2, 10.0.0.3, 10.0.0.4, and 10.0.0.6.

The two routers in AS 65001 each contain one EBGP link to AS 65002 (**R2** and **R4**) over which they announce aggregated prefixes: **100.100/16**. Routers at the edge of a network that communicate directly with routers in other networks are called border routers. Border routers use EBGP to exchange routing information between networks.

Adjacent BGP routers are referred to as neighbors or peers. Peers can be internal or external to the AS. Internal and external peers are configured slightly differently. In general, internal peers communicate using the loopback (**lo0**) interface, and external peers communicate through the shared interface. See [Figure 72 on page 1076](#) for the loopback (**lo0**) and interface information.

To verify the BGP configuration of a router in your network, follow these steps:

## Verify the BGP Configuration

**Purpose** For BGP to run on the router, you must define the local AS number, configure at least one group, and include information about at least one peer in the group (the peer's IP address and AS number). When BGP is part of an MPLS network, you must ensure that the LSP is configured with a destination IP address equal to the BGP next hop in order for BGP routes to be installed with the LSP as the next hop for those routes.

**Action** To verify the BGP configuration, enter the following Junos OS CLI operational mode command:

```
user@host> show configuration
```

## Sample Output 1

```
user@R1> show configuration
[...Output truncated...]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.12.1/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.15.1/30;
      }
      family iso;
      family mpls;
    }
  }
  so-0/0/2 {
    unit 0 {
      family inet {
        address 10.1.13.1/30;
      }
      family iso;
      family mpls;
    }
  }
}
```

```

    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.143/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
            family iso {
                address 49.0004.1000.0000.0001.00;
            }
        }
    }
}
routing-options {
    [...Output truncated...]
    route 100.100.1.0/24 reject;
}
router-id 10.0.0.1;
autonomous-system 65432;
}
protocols {
    rsvp {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path R1-to-R6 {
            to 10.0.0.6; <<< destination address of the LSP
        }
        inactive: interface so-0/0/0.0;
        inactive: interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        export send-statics; <<< missing local-address statement
        group internal {
            type internal;
            neighbor 10.0.0.2;
            neighbor 10.0.0.5;
            neighbor 10.0.0.4;
            neighbor 10.0.0.6;
            neighbor 10.0.0.3;
            neighbor 10.1.36.2; <<< incorrect interface address
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
    }
}

```

```

interface so-0/0/1.0;
interface so-0/0/2.0;
interface all {
    level 2 metric 10;
}
interface fxp0.0 {
    disable;
}
interface lo0.0 {
    passive;
}
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface lo0.0; {
            passive
        }
    }
}
}
policy-options {
    policy-statement send-statics {
        term statics {
            from {
                route-filter 100.100.1.0/24 exact;
            }
            then accept;
        }
    }
}
}

```

## Sample Output 2

```

user@R6> show configuration
[...Output truncated...]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
            family iso;
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.46.2/30;
            }
            family iso;
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {

```

```

        address 10.1.26.2/30;
    }
    family iso;
    family mpls;
}
}
so-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.36.2/30;
        }
        family iso;
        family mpls;
    }
}
fxp0 {
    unit 0 {
        family inet {
            address 192.168.70.148/21;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.6/32;
            address 127.0.0.1/32;
        }
        family iso {
            address 49.0004.1000.0000.0006.00;
        }
    }
}
}
routing-options {
    [...Output truncated...]
    route 100.100.6.0/24 reject;
}
router-id 10.0.0.6;
autonomous-system 65432;
}
protocols {
    rsvp {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface so-0/0/3.0;
        interface fxp0.0 {
            disable;
        }
    }
}
mpls {
    label-switched-path R6-to-R1 {
        to 10.0.0.1; <<< destination address of the reverse LSP
    }
    inactive: interface so-0/0/0.0;
    inactive: interface so-0/0/1.0;
    inactive: interface so-0/0/2.0;
    interface so-0/0/3.0;
}
bgp {

```

```

group internal {
  type internal;
  export send-statics; <<< missing local-address statement
  neighbor 10.0.0.2;
  neighbor 10.0.0.3;
  neighbor 10.0.0.4;
  neighbor 10.0.0.5;
  neighbor 10.0.0.1;
  neighbor 10.1.1.1;          <<< incorrect interface address
}
}
isis {
  level 1 disable;
  interface all {
    level 2 metric 10;
  }
  interface fxp0.0 {
    disable;
  }
  interface lo0.0 {
    passive;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/0/0.0;
    interface so-0/0/1.0;
    interface so-0/0/2.0;
    interface so-0/0/3.0;
    interface lo0.0 {
      passive;
    }
  }
}
}
policy-options {
  policy-statement send-statics {
    term statics {
      from {
        route-filter 100.100.6.0/24 exact;
      }
      then accept;
    }
  }
}
}

```

**Meaning** The sample output shows the BGP configurations on ingress router **R1** and egress router **R6**. Both configurations show the local AS (**65432**), one group (**internal**), and six peers configured. The underlying interior gateway protocol is IS-IS, and the relevant interfaces are configured to run IS-IS.



**NOTE:** In this configuration, the RID is manually configured to avoid any duplicate RID problems, and all interfaces configured with BGP include the family inet statement at the [edit interfaces *type-fpc/pic/port* unit *logical-unit-number*] hierarchy level.

Sample output for ingress router **R1** and egress router **R6** shows that the BGP protocol configuration is missing the **local-address** statement for the internal group. When the **local-address** statement is configured, BGP packets are forwarded from the local router loopback (**lo0**) interface address, which is the address to which BGP peers are peering. If the **local-address** statement is not configured, BGP packets are forwarded from the outgoing interface address, which does not match the address to which BGP peers are peering, and BGP does not come up.

On the ingress router, the IP address (**10.0.0.1**) in the **local-address** statement should be the same as the address configured for the LSP on the egress router (**R6**) in the **to** statement at the **[edit protocols mpls label-switched-path lsp-path-name ]** hierarchy level. BGP uses this address, which is identical to the LSP address, to forward BGP traffic through the LSP.

In addition, the BGP configuration on **R1** includes two IP addresses for **R6**, an interface address (**10.1.36.2**) and a loopback (**lo0**) interface address (**10.0.0.6**), resulting in the LSP destination address (**10.0.0.6**) not matching the BGP next-hop address (**10.1.36.2**). The BGP configuration on **R6** also includes two IP addresses for **R1**, an interface address (**10.1.13.1**) and a loopback (**lo0**) interface address, resulting in the reverse LSP destination address (**10.0.0.1**) not matching the BGP next-hop address (**10.1.13.1**).

In this instance, because the **local-address** statement is missing in the BGP configurations of both routers and the LSP destination address does not match the BGP next-hop address, BGP is not using the LSP to forward traffic.

---

## Display Sent or Received BGP Packets

**Action** To configure the tracing for sent or received BGP protocol packets, follow these steps:

1. In configuration mode, go to the following hierarchy level:

```
[edit]
user@host# edit protocol bgp traceoptions
```

2. Configure the flag to display sent, received, or both sent and received packet information:

```
[edit protocols bgp traceoptions]
user@host# set flag update send
```

or

```
[edit protocols bgp traceoptions]
user@host# set flag update receive
```

or

```
[edit protocols bgp traceoptions]
user@host# set flag update
```

3. Verify the configuration:

```
user@host# show
```

For example:

```
[edit protocols bgp traceoptions]
```

```
user@host# show
file bgplog size 10k files 10;
flag update send;
```

or

```
[edit protocols bgp traceoptions]
user@host# show
file bgplog size 10k files 10;
flag update receive;
```

or

```
[edit protocols bgp traceoptions]
user@host# show
file bgplog size 10k files 10;
flag update send receive;
```

4. Commit the configuration:

```
user@host# commit
```

5. View the contents of the file containing the detailed messages:

```
user@host# run show log filename
```

For example:

```
[edit protocols bgp traceoptions]
user@host# run show log bgplog
Sep 13 12:58:23 trace_on: Tracing to "/var/log/bgplog" started
Sep 13 12:58:23 BGP RECV flags 0x40 code ASPath(2): <null>
Sep 13 12:58:23 BGP RECV flags 0x40 code LocalPref(5): 100
Sep 13 12:58:23 BGP RECV flags 0xc0 code Extended Communities(16): 2:10458:3
[...Output truncated...]
```

## Diagnose BGP Session Establishment Problems

**Purpose** To trace BGP session establishment problems.

**Action** To trace BGP session establishment problems, follow these steps:

1. In configuration mode, go to the following hierarchy level:

```
[edit]
user@host# edit protocol bgp
```

2. Configure BGP open messages:

```
[edit protocols bgp]
user@host# set traceoptions flag open detail
```

3. Verify the configuration:

```
user@host# show
```

For example:

```
[edit protocols bgp]
user@host# show
traceoptions {
```

```
file bgplog size 10k files 10;  
flag open detail;  
}
```

4. Commit the configuration:

```
user@host# commit
```

5. View the contents of the file containing the detailed messages:

```
user@host#run show log filename
```

For example:

```
[edit protocols bgp]
```

```
user@host# run show log bgplog
```

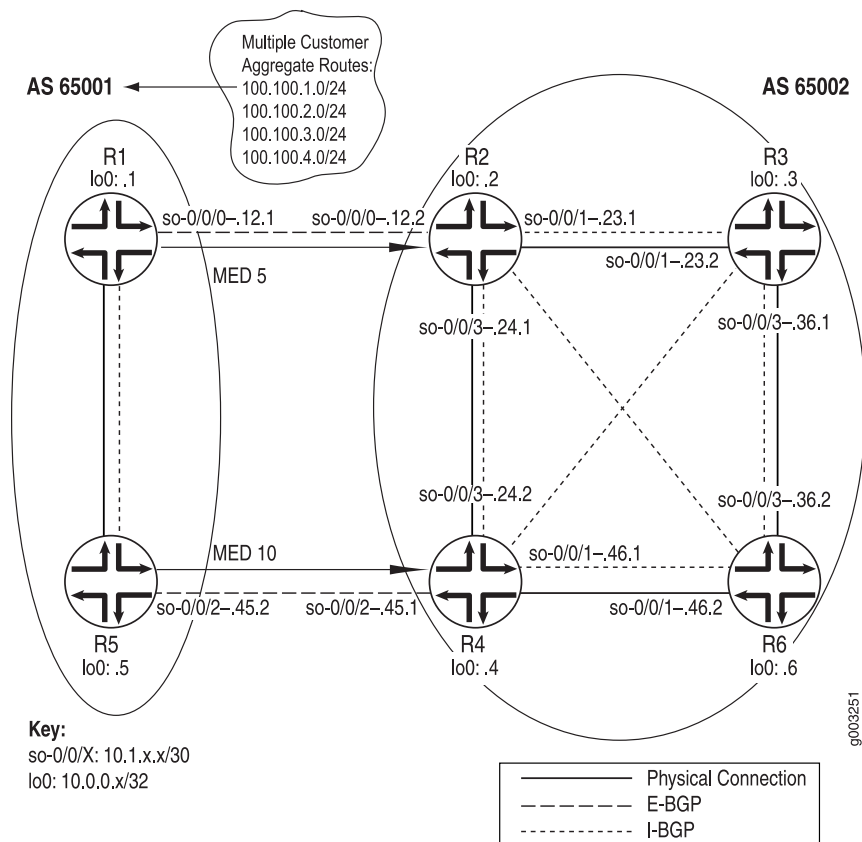
```
Sep 17 17:13:14 trace_on: Tracing to "/var/log/bgplog" started  
Sep 17 17:13:14 bgp_read_v4_update: done with 201.0.0.2 (Internal AS 10458)  
received 19 octets 0 updates 0 routes  
Sep 17 17:13:15 bgp_read_v4_update: receiving packet(s) from 201.0.0.3 (Internal  
AS 10458)  
Sep 17 17:13:15 bgp_read_v4_update: done with 201.0.0.3 (Internal AS 10458)  
received 19 octets 0 updates 0 routes  
Sep 17 17:13:44 bgp_read_v4_update: receiving packet(s) from 201.0.0.2 (Internal  
AS 10458)  
[...Output truncated...]
```

---

## Examine BGP Routes and Route Selection

**Purpose** You can examine the BGP path selection process to determine the single, active path when BGP receives multiple routes to the same destination prefix.

Figure 73: BGP Network Topology



The network in [Figure 73 on page 1085](#) shows that **R1** and **R5** announce the same aggregate routes to **R2** and **R4**, which results in **R2** and **R4** receiving two routes to the same destination prefix. The route selection process on **R2** and **R4** determines which of the two BGP routes received is active and advertised to the other internal routers (**R3** and **R6**).

Before the router installs a BGP route, it must make sure that the BGP **next-hop** attribute is reachable. If the BGP next hop cannot be resolved, the route is not installed. When a BGP route is installed in the routing table, it must go through a path selection process if multiple routes exist to the same destination prefix. The BGP path selection process proceeds in the following order:

1. Route preference in the routing table is compared. For example, if an OSPF and a BGP route exist for a particular destination, the OSPF route is selected as the active route because the OSPF route has a default preference of 110, while the BGP route has a default preference of 170.
2. Routes are compared for local preference. The route with the highest local preference is preferred. For example, see [“Examine the Local Preference Selection” on page 1086](#).
3. The AS path attribute is evaluated. The shorter AS path is preferred.
4. The origin code is evaluated. The lowest origin code is preferred ( **I** (IGP) < **E** (EGP) < **?** (Incomplete)).

5. The MED value is evaluated. By default, if any of the routes are advertised from the same neighboring AS, the lowest MED value is preferred. The absence of a MED value is interpreted as a MED of 0. For an example, see [“Examine the Multiple Exit Discriminator Route Selection” on page 1087](#).
6. The route is evaluated as to whether it is learned through EBGp or IBGP. EBGp learned routes are preferred to IBGP learned routes. For an example, see [“Examine the EBGp over IBGP Selection” on page 1088](#).
7. If the route is learned from IBGP, the route with the lowest IGP cost is preferred. For an example, see [“Examine the IGP Cost Selection” on page 1089](#). The physical next hop to the IBGP peer is installed according to the following three rules:
  - a. After BGP examines the **inet.0** and **inet.3** routing tables, the physical next hop of the route with the lowest preference is used.
  - b. If the preference values in the **inet.0** and the **inet.3** routing tables are a tie, the physical next hop of the route in the **inet.3** routing table is used.
  - c. When a preference tie exists in the same routing table, the physical next hop of the route with more paths is installed.
8. The route reflection cluster list attribute is evaluated. The shortest length cluster list is preferred. Routes without a cluster list are considered to have a cluster list length of 0.
9. The router ID is evaluated. The route from the peer with the lowest router ID is preferred (usually the loopback address).
10. The peer address value is examined. The peer with the lowest peer IP address is preferred.

To determine the single, active path when BGP receives multiple routes to the same destination prefix, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix < detail >
```

The following steps illustrate the inactive reason displayed when BGP receives multiple routes to the same destination prefix and one route is selected as the single, active path:

1. [Examine the Local Preference Selection on page 1086](#)
2. [Examine the Multiple Exit Discriminator Route Selection on page 1087](#)
3. [Examine the EBGp over IBGP Selection on page 1088](#)
4. [Examine the IGP Cost Selection on page 1089](#)

## Examine the Local Preference Selection

- |                |   |
|----------------|---|
| <b>Purpose</b> | To examine a route to determine if local preference is the selection criteria for the single, active path.  |
| <b>Action</b>  | To examine a route to determine if local preference is the selection criteria for the single, active path, enter the following Junos OS CLI operational mode command: |

```
user@host> show route destination-prefix < detail >
```

## Sample Output

```
user@R4> show route 100.100.1.0 detail
inet.0: 20 destinations, 24 routes (20 active, 0 holddown, 0 hidden)
100.100.1.0/24 (2 entries, 1 announced)
    *BGP      Preference: 170/-201
      Source: 10.0.0.2
      Next hop: 10.1.24.1 via so-0/0/3.0, selected
      Protocol next hop: 10.0.0.2 Indirect next hop: 8644000 277
      State: <Active Int Ext>
      Local AS: 65002 Peer AS: 65002
      Age: 2:22:34      Metric: 5      Metric2: 10
      Task: BGP_65002.10.0.0.2+179
      Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0

      AS path: 65001|
      Localpref: 200
      Router ID: 10.0.0.2
    BGP      Preference: 170/-101
      Source: 10.1.45.2
      Next hop: 10.1.45.2 via so-0/0/2.0, selected
      State: <Ext>
      Inactive reason: Local Preference
      Local AS: 65002 Peer AS: 65001
      Age: 2w0d 1:28:31      Metric: 10
      Task: BGP_65001.10.1.45.2+179
      AS path: 65001|
      Localpref: 100
      Router ID: 10.0.0.5
```

**Meaning** The sample output shows that R4 received two instances of the 100.100.1.0 route: one from 10.0.0.2 (R2) and one from 10.1.45.2 (R5). R4 selected the path from R2 as its active path, as indicated by the asterisk (\*). The selection is based on the local preference value contained in the **Localpref** field. The path with the *highest* local preference is preferred. In the example, the path with the higher local preference value is the path from R2, 200.

The reason that the route from R5 is not selected is in the **Inactive reason** field, in this case, **Local Preference**.

Note that the two paths are from the same neighboring network: AS 65001.

## Examine the Multiple Exit Discriminator Route Selection

**Purpose** To examine a route to determine if the MED is the selection criteria for the single, active path.

**Action** To examine a route to determine if the MED is the selection criteria for the single, active path, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix < detail >
```

## Sample Output

```
user@R4> show route 100.100.2.0 detail
```

```

inet.0: 20 destinations, 24 routes (20 active, 0 holddown, 0 hidden)
100.100.2.0/24 (2 entries, 1 announced)
  *BGP      Preference: 170/-101
    Source: 10.0.0.2
    Next hop: 10.1.24.1 via so-0/0/3.0, selected
    Protocol next hop: 10.0.0.2 Indirect next hop: 8644000 277
    State: <Active Int Ext>
    Local AS: 65002 Peer AS: 65002
    Age: 2:32:01      Metric: 5      Metric2: 10
    Task: BGP_65002.10.0.0.2+179
    Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0

    AS path: 65001|
    Localpref: 100
    Router ID: 10.0.0.2
  BGP      Preference: 170/-101
    Source: 10.1.45.2
    Next hop: 10.1.45.2 via so-0/0/2.0, selected
    State: <NotBest Ext>
    Inactive reason: Not Best in its group
    Local AS: 65002 Peer AS: 65001
    Age: 2w0d 1:37:58      Metric: 10
    Task: BGP_65001.10.1.45.2+179
    AS path: 65001|
    Localpref: 100
    Router ID: 10.0.0.5

```

**Meaning** The sample output shows that R4 received two instances of the 100.100.2.0 route: one from 10.0.0.2 (R2), and one from 10.1.45.2 (R5). R4 selected the path from R2 as its active route, as indicated by the asterisk (\*). The selection is based on the MED value contained in the **Metric** field. The path with the lowest MED value is preferred. In the example, the path with the lowest MED value (5) is the path from R2. Note that the two paths are from the same neighboring network: AS 65001.

The reason that the inactive path is not selected is displayed in the **Inactive reason** field, in this case, **Not Best in its group**. The wording is used because the Junos OS uses the process of deterministic MED selection, by default.

## Examine the EBGP over IBGP Selection

**Purpose** To examine a route to determine if EBGP is selected over IBGP as the selection criteria for the single, active path.

**Action** To examine a route to determine if EBGP is selected over IBGP as the selection criteria for the single, active path, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix < detail >
```

## Sample Output

```

user@R4> show route 100.100.3.0 detail
inet.0: 20 destinations, 24 routes (20 active, 0 holddown, 0 hidden)
100.100.3.0/24 (2 entries, 1 announced)
  *BGP      Preference: 170/-101
    Source: 10.1.45.2
    Next hop: 10.1.45.2 via so-0/0/2.0, selected

```

```

State: <Active Ext>
  Local AS: 65002 Peer AS: 65001
Age: 5d 0:31:25
Task: BGP_65001.10.1.45.2+179
Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0

AS path: 65001 I
Localpref: 100
Router ID: 10.0.0.5
BGP Preference: 170/-101
  Source: 10.0.0.2
Next hop: 10.1.24.1 via so-0/0/3.0, selected
Protocol next hop: 10.0.0.2 Indirect next hop: 8644000 277
State: <NotBest Int Ext>
  Inactive reason: Interior > Exterior > Exterior via Interior
  Local AS: 65002 Peer AS: 65002
Age: 2:48:18 Metric2: 10
Task: BGP_65002.10.0.0.2+179
AS path: 65001 I
Localpref: 100
Router ID: 10.0.0.2

```

**Meaning** The sample output shows that **R4** received two instances of the **100.100.3.0** route: one from **10.1.45.2 (R5)** and one from **10.0.0.2 (R2)**. **R4** selected the path from **R5** as its active path, as indicated by the asterisk (\*). The selection is based on a preference for routes learned from an EBGP peer over routes learned from an IBGP. **R5** is an EBGP peer.

You can determine if a path is received from an EBGP or IBGP peer by examining the **Local As** and **Peer As** fields. For example, the route from **R5** shows the local AS is 65002 and the peer AS is 65001, indicating that the route is received from an EBGP peer. The route from **R2** shows that both the local and peer AS is 65002, indicating that it is received from an IBGP peer.

The reason that the inactive path is not selected is displayed in the **Inactive reason** field, in this case, **Interior > Exterior > Exterior via Interior**. The wording of this reason shows the order of preferences applied when the same route is received from two routers. The route received from a strictly internal source (IGP) is preferred first, the route received from an external source (EBGP) is preferred next, and any route which comes from an external source and is received internally (IBGP) is preferred last.

## Examine the IGP Cost Selection

**Purpose** To examine a route to determine if EBGP is selected over IBGP as the selection criteria for the single, active path.

**Action** To examine a route to determine if EBGP is selected over IBGP as the selection criteria for the single, active path, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix < detail >
```

## Sample Output

```

user@R6> show route 100.100.4.0 detail
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
100.100.4.0/24 (2 entries, 1 announced)

```

```

*BGP      Preference: 170/-101
          Source: 10.0.0.4
          Next hop: 10.1.46.1 via so-0/0/1.0, selected
          Protocol next hop: 10.0.0.4 Indirect next hop: 864c000 276
          State: <Active Int Ext>
          Local AS: 65002 Peer AS: 65002
          Age: 2:16:11      Metric2: 10
          Task: BGP_65002.10.0.0.4+4120
          Announcement bits (2): 0-KRT 4-Resolve inet.0
          AS path: 65001|
          Localpref: 100
          Router ID: 10.0.0.4
BGP      Preference: 170/-101
          Source: 10.0.0.2
          Next hop: 10.1.46.1 via so-0/0/1.0, selected
          Next hop: 10.1.36.1 via so-0/0/3.0
          Protocol next hop: 10.0.0.2 Indirect next hop: 864c0b0 278
          State: <NotBest Int Ext>
          Inactive reason: IGP metric
          Local AS: 65002 Peer AS: 65002
          Age: 2:16:03      Metric2: 20
          Task: BGP_65002.10.0.0.2+179
          AS path: 65001|
          Localpref: 100
          Router ID: 10.0.0.2

```

**Meaning** The sample output shows that **R6** received two instances of the **100.100.4.0** route: one from **10.0.0.4 (R4)** and one from **10.0.0.2 (R2)**. **R6** selected the path from **R4** as its active route, as indicated by the asterisk (\*). The selection is based on the IGP metric, displayed in the **Metric2** field. The route with the lowest IGP metric is preferred. In the example, the path with the lowest IGP metric value is the path from **R4**, with an IGP metric value of 10, while the path from **R2** has an IGP metric of 20. Note that the two paths are from the same neighboring network: AS 65001.

The reason that the inactive path was not selected is displayed in the **Inactive reason** field, in this case, **IGP metric**.

## Examine Routes in the Forwarding Table

**Purpose** When you run into problems, such as connectivity problems, you may need to examine routes in the forwarding table to verify that the routing protocol process has relayed the correct information into the forwarding table.

**Action** To display the set of routes installed in the forwarding table, enter the following Junos OS CLI operational mode command:

```
user@host> show route forwarding-table
```

## Sample Output

```

user@R2> show route forwarding-table
Routing table: inet
Internet:

```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		reject	10	1	
10.0.0.2/32	intf	0	10.0.0.2	local	256	1	

```

10.0.0.3/32      user      1 10.1.23.0      ucst  282    4 so-0/0/1.0
10.0.0.4/32      user      1 10.1.24.0      ucst  290    7 so-0/0/3.0
10.0.0.6/32      user      1 10.1.24.0      ucst  290    7 so-0/0/3.0
10.1.12.0/30     intf      1 ff.3.0.21      ucst  278    6 so-0/0/0.0
10.1.12.0/32     dest      0 10.1.12.0      recv  280    1 so-0/0/0.0
10.1.12.2/32     intf      0 10.1.12.2      locl   277    1
10.1.12.3/32     dest      0 10.1.12.3      bcst   279    1 so-0/0/0.0
10.1.23.0/30     intf      0 ff.3.0.21      ucst  282    4 so-0/0/1.0
10.1.23.0/32     dest      0 10.1.23.0      recv  284    1 so-0/0/1.0
10.1.23.1/32     intf      0 10.1.23.1      locl   281    1
10.1.23.3/32     dest      0 10.1.23.3      bcst   283    1 so-0/0/1.0
10.1.24.0/30     intf      0 ff.3.0.21      ucst  290    7 so-0/0/3.0
10.1.24.0/32     dest      0 10.1.24.0      recv  292    1 so-0/0/3.0
10.1.24.1/32     intf      0 10.1.24.1      locl   289    1
10.1.24.3/32     dest      0 10.1.24.3      bcst   291    1 so-0/0/3.0
10.1.36.0/30     user      0 10.1.23.0      ucst  282    4 so-0/0/1.0
10.1.46.0/30     user      0 10.1.24.0      ucst  290    7 so-0/0/3.0
100.100.1.0/24   user      0 10.1.12.0      ucst  278    6 so-0/0/0.0
100.100.2.0/24   user      0 10.1.12.0      ucst  278    6 so-0/0/0.0
100.100.3.0/24   user      0 10.1.12.0      ucst  278    6 so-0/0/0.0
100.100.4.0/24   user      0 10.1.12.0      ucst  278    6 so-0/0/0.0
[...Output truncated...]

```

**Meaning** The sample output shows the network-layer prefixes and their next hops installed in the forwarding table. The output includes the same next-hop information as in the **show route detail** command (the next-hop address and interface name). Additional information includes the destination type, the next-hop type, the number of references to this next hop, and an index into an internal next-hop database. (The internal database contains additional information used by the Packet Forwarding Engine to ensure proper encapsulation of packets sent out an interface. This database is not accessible to the user.

For detailed information about the meanings of the various flags and types fields, see the *Junos Routing Protocols and Policies Command Reference*.

## Log BGP State Transition Events

**Purpose** Border Gateway Protocol (BGP) state transitions indicate a network problem and need to be logged and investigated.

**Action** To log BGP state transition events to the system log, follow these steps:

1. In configuration mode, go to the following hierarchy level:

```
[edit]
user@host# edit protocol bgp
```

2. Configure the system log:

```
user@host# set log-updown
```

3. Verify the configuration:

```
user@host# show
```

4. Commit the configuration:

```
user@host# commit
```

**Meaning** Log messages from BGP state transition events are sufficient to diagnose most BGP session problems. [Table 46 on page 1092](#) lists and describes the six states of a BGP session.

**Table 46: Six States of a BGP Session**

BGP State	Description
<b>Idle</b>	<p>This is the first state of a connection. BGP waits for a start event initiated by an administrator. The start event might be the establishment of a BGP session through router configuration or the resetting of an existing session. After the start event, BGP initializes its resources, resets a connect-retry timer, initiates a TCP transport connection, and starts listening for connections initiated by remote peers. BGP then transitions to a <b>Connect</b> state.</p> <p>If there are errors, BGP falls back to the <b>Idle</b> state.</p>
<b>Connect</b>	<p>BGP waits for the transport protocol connection to complete. If the TCP transport connection is successful, the state transitions to <b>OpenSent</b>.</p> <p>If the transport connection is not successful, the state transitions to <b>Active</b>.</p> <p>If the connect-retry timer has expired, the state remains in the <b>Connect</b> state, the timer is reset, and a transport connection is initiated.</p> <p>With any other event, the state goes back to <b>Idle</b>.</p>
<b>Active</b>	<p>BGP tries to acquire a peer by initiating a transport protocol connection.</p> <p>If it is successful, the state transitions to <b>OpenSent</b>.</p> <p>If the connect-retry timer expires, BGP restarts the connect timer and falls back to the <b>Connect</b> state. BGP continues to listen for a connection that may be initiated from another peer. The state may go back to <b>Idle</b> in case of other events, such as a stop event.</p> <p>In general, a neighbor state flip-flopping between <b>Connect</b> and <b>Active</b> is an indication that there is a problem with the TCP transport connection. Such a problem might be caused by many TCP retransmissions or the inability of a neighbor to reach the IP address of its peer.</p>
<b>OpenSent</b>	<p>BGP receives an open message from its peer. In the <b>OpenSent</b> state, BGP compares its autonomous system (AS) number with the AS number of its peer and recognizes whether the peer belongs to the same AS (internal BGP) or to a different AS (external BGP).</p> <p>The open message is checked for correctness. In case of errors, such as a bad version number of an unacceptable AS, BGP sends an error-notification message and goes back to <b>Idle</b>.</p> <p>For any other errors, such as expiration of the hold timer or a stop event, BGP sends a notification message with the corresponding error code and falls back to the <b>Idle</b> state.</p> <p>If there are no errors, BGP sends keepalive messages and resets the keepalive timer. In this state, the hold time is negotiated. If the hold time is 0, the hold and keepalive timers are not restarted.</p> <p>When a TCP transport disconnect is detected, the state falls back to <b>Active</b>.</p>

Table 46: Six States of a BGP Session (*continued*)

BGP State	Description
<b>OpenConfirm</b>	<p>BGP waits for a keepalive or notification message.</p> <p>If a keepalive is received, the state becomes <b>Established</b>, and the neighbor negotiation is complete. If the system receives an update or keepalive message, it restarts the hold timer (assuming that the negotiated hold time is not 0).</p> <p>If a notification message is received, the state falls back to <b>Idle</b>.</p> <p>The system sends periodic keepalive messages at the rate set by the keepalive timer. In case of a transport disconnect notification or in response to a stop event, the state falls back to <b>Idle</b>. In response to other events, the system sends a notification message with a finite state machine (FSM) error code and goes back to <b>Idle</b>.</p>
<b>Established</b>	<p>This is the final state in the neighbor negotiation. In this state, BGP exchanges update packets with its peers and the hold timer is restarted at the receipt of an update or keepalive message when it is not set to zero.</p> <p>If the system receives a notification message, the state falls back to <b>Idle</b>.</p> <p>Update messages are checked for errors, such as missing attributes, duplicate attributes, and so on. If errors are found, a notification is sent to the peer, and the state falls back to <b>Idle</b>.</p> <p>BGP goes back to <b>Idle</b> when the hold timer expires, a disconnect notification is received from the transport protocol, a stop event is received, or in response to any other event.</p>

For more detailed BGP protocol packet information, configure BGP-specific tracing. See *Checklist for Tracking Error Conditions* for more information.

## Display Detailed BGP Protocol Information

**Action** To display BGP protocol information in detail, follow these steps:

1. In configuration mode, go to the following hierarchy level:

```
[edit]
user@host# edit protocol bgp traceoptions
```

2. Configure the flag to display detailed BGP protocol messages:

```
[edit protocols bgp traceoptions]
user@host# set flag update detail
```

3. Verify the configuration:

```
user@host# show
```

For example:

```
[edit protocols bgp traceoptions]
user@host# show
flag update detail;
```

- Commit the configuration:

```
user@host# commit
```

- View the contents of the file containing the detailed messages:

```
user@host# run show log filename
```

For example:

```
[edit protocols bgp traceoptions]
user@pro5-a# run show log bgp
Sep 17 14:47:16 trace_on: Tracing to "/var/log/bgp" started
Sep 17 14:47:17 bgp_read_v4_update: receiving packet(s) from 10.255.245.53 (Internal
AS 10458)
Sep 17 14:47:17 BGP RECV 10.255.245.53+179 -> 10.255.245.50+1141
Sep 17 14:47:17 BGP RECV message type 2 (Update) length 128
Sep 17 14:47:17 BGP RECV flags 0x40 code Origin(1): IGP
Sep 17 14:47:17 BGP RECV flags 0x40 code ASPath(2): 2
Sep 17 14:47:17 BGP RECV flags 0x80 code MultiExitDisc(4): 0
Sep 17 14:47:17 BGP RECV flags 0x40 code LocalPref(5): 100
Sep 17 14:47:17 BGP RECV flags 0xc0 code Extended Communities(16): 2:10458:1
[...Output truncated...]
```

**Meaning** [Table 47 on page 1094](#) lists tracing flags specific to BGP and presents example output for some of the flags. You can also configure tracing for a specific BGP peer or peer group. For more information, see the *Junos System Basics Configuration Guide*.

**Table 47: BGP Protocol Tracing Flags**

Tracing Flags	Description	Example Output
<b>aspath</b>	AS path regular expression operations	Not available.
<b>damping</b>	Damping operations	Nov 28 17:01:12 bgp_damp_change: Change event Nov 28 17:01:12 bgp_dampen: Damping 10.10.1.0 Nov 28 17:01:12 bgp_damp_change: Change event Nov 28 17:01:12 bgp_dampen: Damping 10.10.2.0 Nov 28 17:01:12 bgp_damp_change: Change event Nov 28 17:01:12 bgp_dampen: Damping 10.10.3.0
<b>keepalive</b>	BGP keepalive messages	Nov 28 17:09:27 bgp_send: sending 19 bytes to 10.217.5.101 (External AS 65471) Nov 28 17:09:27 Nov 28 17:09:27 BGP SEND 10.217.5.1+179 -> 10.217.5.101+52162 Nov 28 17:09:27 BGP SEND message type 4 (KeepAlive) length 19 Nov 28 17:09:28 Nov 28 17:09:28 BGP RECV 10.217.5.101+52162 -> 10.217.5.1+179 Nov 28 17:09:28 BGP RECV message type 4 (KeepAlive) length 19
<b>open</b>	BGP open packets	Nov 28 18:37:42 bgp_send: sending 37 bytes to 10.217.5.101 (External AS 65471) Nov 28 18:37:42 Nov 28 18:37:42 BGP SEND 10.217.5.1+179 -> 10.217.5.101+38135 Nov 28 18:37:42 BGP SEND message type 1 (Open) length 37

Table 47: BGP Protocol Tracing Flags (*continued*)

Tracing Flags	Description	Example Output
<b>packets</b>	All BGP protocol packets	<pre>Sep 27 17:45:31 BGP RECV 10.0.100.108+179 -&gt; 10.0.100.105+1033 Sep 27 17:45:31 BGP RECV message type 4 (KeepAlive) length 19 Sep 27 17:45:31 bgp_send: sending 19 bytes to 10.0.100.108 (Internal AS 100) Sep 27 17:45:31 BGP SEND 10.0.100.105+1033 -&gt; 10.0.100.108+179 Sep 27 17:45:31 BGP SEND message type 4 (KeepAlive) length 19 Sep 27 17:45:31 bgp_read_v4_update: receiving packet(s) from 10.0.100.108 (Internal AS 100)</pre>
<b>update</b>	Update packets	<pre>Nov 28 19:05:24 BGP SEND 10.217.5.1+179 -&gt; 10.217.5.101+55813 Nov 28 19:05:24 BGP SEND message type 2 (Update) length 53 Nov 28 19:05:24 bgp_send: sending 65 bytes to 10.217.5.101 (External AS 65471) Nov 28 19:05:24 Nov 28 19:05:24 BGP SEND 10.217.5.1+179 -&gt; 10.217.5.101+55813 Nov 28 19:05:24 BGP SEND message type 2 (Update) length 65 Nov 28 19:05:24 bgp_send: sending 55 bytes to 10.217.5.101 (External AS 65471)</pre>

## Verify Received BGP Routes

**Purpose** Display the routing information received on router **R6**, the ingress router for the reverse LSP **R6-to-R1**.

**Action** To verify that a particular BGP route is received on the egress router, enter the following Junos OS CLI operational mode command:

```
user@host> show route receive protocol bgp neighbor-address
```

### Sample Output 1

```
user@R6> show route receive-protocol bgp 10.0.0.1
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
<<< missing route
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: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

### Sample Output 2

```
user@R6> show route receive-protocol bgp 10.0.0.1
inet.0: 30 destinations, 46 routes (29 active, 0 holddown, 1 hidden)
  Prefix                Nexthop              MED      Lc1pref  AS path
*100.100.1.0/24    10.0.0.1             100      I

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: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

```
__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

**Meaning** Sample Output 1 shows that ingress router **R6** (reverse LSP **R6-to-R1**) does not receive any BGP routes into the **inet.0** routing table when the BGP configurations of **R1** and **R6** are incorrect.

Sample Output 2 shows a BGP route installed in the **inet.0** routing table after the BGP configurations on **R1** and **R6** are corrected using [“Take Appropriate Action” on page 1070](#).

## Verify That a Particular BGP Route Is Received on Your Router

**Purpose** Display the routing information as it is received through a particular BGP neighbor and advertised by the local router to the neighbor.

**Action** To verify that a particular BGP route is received on your router, enter the following Junos OS CLI operational mode command:

```
user@host> show route receive-protocol bgp neighbor-address
```

### Sample Output

```
user@R6> show route receive-protocol bgp 10.0.0.2
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref      AS path
*100.100.1.0/24    10.0.0.2         5      200      65001 I
*100.100.2.0/24    10.0.0.2         5      100      65001 I
  100.100.3.0/24    10.0.0.2                100          65001 I
  100.100.4.0/24    10.0.0.2                100          65001 I
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
user@R6> show route receive-protocol bgp 10.0.0.4
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lc1pref      AS path
*100.100.3.0/24    10.0.0.4                100          65001 I
*100.100.4.0/24    10.0.0.4                100          65001 I
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

**Meaning** The sample output shows four BGP routes from **R2** and two from **R4**. Of the four routes from **R2**, only two are active in the routing table, as indicated by the asterisk (\*), while both routes received from **R4** are active in the routing table. All BGP routes came through AS 65001.

## Verifying Advertised BGP Routes

**Purpose** You can determine if a particular route that you have configured is being advertised to a neighbor.

**Action** To verify the routing information as it has been prepared for advertisement to the specified BGP neighbor, enter the following Junos OS CLI operational mode command:

```
user@host> show route advertising-protocol bgp neighbor-address
```

## Sample Output

```
user@R2> show route advertising-protocol bgp 10.0.0.4\
inet.0: 20 destinations, 22 routes (20 active, 0 hold-down, 0 hidden)
  Prefix                Nexthop          MED      Lc1pref   AS path
* 100.100.1.0/24        Self             5         200       65001 I
* 100.100.2.0/24        Self             5         100       65001 I
* 100.100.3.0/24        Self             100       65001 I
* 100.100.4.0/24        Self             100       65001 I
```

**Meaning** The sample output shows the BGP routes advertised from **R2** to its neighbor, **10.0.0.4 (R4)**. Out of 22 total routes in the **inet.0** routing table, 20 are active destinations. No routes are **hidden** or in the **hold-down** state. Routes reside in the **hold-down** state prior to being declared active, and routes rejected by a routing policy can be placed into the **hidden** state. The information displayed reflects the routes that the routing table exported to the BGP routing protocol.

## Check That BGP Traffic Is Using the LSP

**Purpose** At this level of the troubleshooting model, BGP and the LSP may be up, however BGP traffic might not be using the LSP to forward traffic.

**Action** To verify that BGP traffic is using the LSP, enter the following Junos OS command-line interface (CLI) operational mode command from the ingress router:

```
user@host> traceroute hostname
```

## Sample Output

```
user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1  10.1.13.2 (10.1.13.2)  0.653 ms  0.590 ms  0.543 ms
 2  10.1.36.2 (10.1.36.2)  0.553 ms !N  0.552 ms !N  0.537 ms !N

user@R6> traceroute 100.100.1.1
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
 1  10.1.36.1 (10.1.36.1)  0.660 ms  0.551 ms  0.526 ms
 2  10.1.13.1 (10.1.13.1)  0.568 ms !N  0.553 ms !N  0.536 ms !N
```

**Meaning** The sample output shows that BGP traffic is not using the LSP, consequently MPLS labels do not appear in the output. Instead of using the LSP, BGP traffic is using the interior gateway protocol (IGP) to reach the BGP next-hop LSP egress address for **R6** and **R1**. The Junos OS default is to use LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

## Check That BGP Traffic Is Using the LSP Again

**Purpose** After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that BGP traffic is using the LSP and that the problem in the BGP layer has been resolved.

**Action** To verify that BGP traffic is using the LSP, enter the following Junos OS CLI operational mode command from the ingress router:

```
user@host> traceroute hostname
```

## Sample Output

```
user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1 10.1.13.2 (10.1.13.2) 0.858 ms 0.740 ms 0.714 ms
    MPLS Label=100016 CoS=0 TTL=1 S=1
 2 10.1.36.2 (10.1.36.2) 0.592 ms !N 0.564 ms !N 0.548 ms !N

user@R6> traceroute 100.100.1.1
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
 1 10.1.36.1 (10.1.36.1) 0.817 ms 0.697 ms 0.771 ms
    MPLS Label=100000 CoS=0 TTL=1 S=1
 2 10.1.13.1 (10.1.13.1) 0.581 ms !N 0.567 ms !N 0.544 ms !N
```

**Meaning** The sample output shows that MPLS labels are used to forward packets through the LSP. Included in the output is a label value (**MPLS Label=100016**), the time-to-live value (**TTL=1**), and the stack bit value (**S=1**).

The **MPLS Label** field is used to identify the packet to a particular LSP. It is a 20-bit field, with a maximum value of ( $2^{20}-1$ ), approximately 1,000,000.

The time-to-live (TTL) value contains a limit on the number of hops that this MPLS packet can travel through the network (1). It is decremented at each hop, and if the TTL value drops below one, the packet is discarded.

The bottom of the stack bit value (**S=1**) indicates that is the last label in the stack and that this MPLS packet has one label associated with it. The MPLS implementation in the Junos OS supports a stacking depth of 3 on the M-series routers and up to 5 on the T-series routing platforms. For more information on MPLS label stacking, see RFC 3032, *MPLS Label Stack Encoding*.

MPLS labels appear in the sample output because the **traceroute** command is issued to a BGP destination where the BGP next hop for that route is the LSP egress address. The Junos OS by default uses LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

If the BGP next hop does not equal the LSP egress address, the BGP traffic does not use the LSP, and consequently MPLS labels do not appear in the output for the **traceroute** command, as indicated in the sample output in [“Check BGP Sessions” on page 1061](#).

## Examine the EBGp over IBGP Selection

**Purpose** To examine a route to determine if EBGp is selected over IBGP as the selection criteria for the single, active path.

**Action** To examine a route to determine if EBGp is selected over IBGP as the selection criteria for the single, active path, enter the following Junos OS CLI operational mode command:

```
user@host> show route destination-prefix < detail >
```

## Sample Output

```

user@R4> show route 100.100.3.0 detail
inet.0: 20 destinations, 24 routes (20 active, 0 holddown, 0 hidden)
100.100.3.0/24 (2 entries, 1 announced)
    *BGP      Preference: 170/-101
        Source: 10.1.45.2
        Next hop: 10.1.45.2 via so-0/0/2.0, selected
        State: <Active Ext>
        Local AS: 65002 Peer AS: 65001
        Age: 5d 0:31:25
        Task: BGP_65001.10.1.45.2+179
        Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-Resolve inet.0

    BGP      Preference: 170/-101
        Source: 10.0.0.2
        Next hop: 10.1.24.1 via so-0/0/3.0, selected
        Protocol next hop: 10.0.0.2 Indirect next hop: 8644000 277
        State: <NotBest Int Ext>
        Inactive reason: Interior > Exterior > Exterior via Interior
        Local AS: 65002 Peer AS: 65002
        Age: 2:48:18    Metric2: 10
        Task: BGP_65002.10.0.0.2+179
        AS path: 65001 I
        Localpref: 100
        Router ID: 10.0.0.2

```

**Meaning** The sample output shows that **R4** received two instances of the **100.100.3.0** route: one from **10.1.45.2 (R5)** and one from **10.0.0.2 (R2)**. **R4** selected the path from **R5** as its active path, as indicated by the asterisk (\*). The selection is based on a preference for routes learned from an EBGp peer over routes learned from an IBGP. **R5** is an EBGp peer.

You can determine if a path is received from an EBGp or IBGP peer by examining the **Local As** and **Peer As** fields. For example, the route from **R5** shows the local AS is 65002 and the peer AS is 65001, indicating that the route is received from an EBGp peer. The route from **R2** shows that both the local and peer AS is 65002, indicating that it is received from an IBGP peer.

The reason that the inactive path is not selected is displayed in the **Inactive reason** field, in this case, **Interior > Exterior > Exterior via Interior**. The wording of this reason shows the order of preferences applied when the same route is received from two routers. The route received from a strictly internal source (IGP) is preferred first, the route received from an external source (EBGP) is preferred next, and any route which comes from an external source and is received internally (IBGP) is preferred last.

## Verify BGP on an Internal Router

**Purpose** To verify the BGP configuration of an internal router.

**Action** To verify the BGP configuration of an internal router, enter the following Junos OS command-line interface (CLI) command:

```
user@host> show configuration
```

The following sample output is for a BGP configuration on R3, as shown in [“Verify the BGP Protocol” on page 1075](#):

## Sample Output

```
user@R3> show configuration
[...Output truncated...]
interfaces {
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.23.2/30;
      }
      family iso;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.1.36.1/30;
      }
      family iso;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.3/32;
      }
      family iso {
        address 49.0002.1000.0000.0003.00;
      }
    }
  }
}
routing-options {
  [...Output truncated...]
  router-id 10.0.0.3;
  autonomous-system 65002;
}
protocols {
  bgp {
    group internal {
      type internal;
      local-address 10.0.0.3;
      neighbor 10.0.0.2;
      neighbor 10.0.0.4;
      neighbor 10.0.0.6;
    }
  }
  isis {
    level 1 disable;
    interface all {
      level 2 metric 10;
    }
    interface lo0.0;
  }
}
```

```

user@R6> show configuration |
[Output truncated...]
interfaces {
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.46.2/30;
      }
      family iso;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.1.36.2/30;
      }
      family iso;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.6/32;
      }
      family iso {
        address 49.0003.1000.0000.0006.00;
      }
    }
  }
}
routing-options {
  [Output truncated...]
  router-id 10.0.0.6;
  autonomous-system 65002;
}
protocols {
  bgp {
    group internal {
      type internal;
      local-address 10.0.0.6;
      neighbor 10.0.0.2;
      neighbor 10.0.0.3;
      neighbor 10.0.0.4;
    }
  }
  isis {
    level 1 disable;
    interface all {
      level 2 metric 10;
    }
    interface lo0.0;
  }
}

```

**Meaning** The sample output shows a basic BGP configuration on routers **R3** and **R6**. The local AS (65002) and one group (**internal**) are configured on both routers. **R3** has three internal peers—**10.0.0.2**, **10.0.0.4**, and **10.0.0.6**—included at the [**protocols bgp group group**] hierarchy level. **R6** also has three internal peers: **10.0.0.2**, **10.0.0.3**, and **10.0.0.4**. The

underlying IGP protocol is Intermediate System-to-Intermediate System (IS-IS), and relevant interfaces are configured to run IS-IS.

Note that in this configuration the router ID is manually configured to avoid any duplicate router ID problems.

---

## Verify BGP on a Border Router

**Purpose** To verify the BGP configuration of a border router.

**Action** To verify the BGP configuration of a border router, enter the following Junos OS CLI operational mode command:

```
user@host> show configuration
```

### Sample Output

The following sample output is for a BGP configuration on two border routers from AS 65002 (R2 and R4 as shown in ["Verify the BGP Protocol" on page 1075](#)):

```
user@R2> show configuration
[...Output truncated...]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.12.2/30;
      }
      family iso;
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.23.1/30;
      }
      family iso;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.1.24.1/30;
      }
      family iso;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.2/32;
      }
      family iso {
        address 49.0002.1000.0000.0002.00;
      }
    }
  }
}
```

```

routing-options {
[...Output truncated...]
  router-id 10.0.0.2;
  autonomous-system 65002;
}
protocols {
  bgp {
    group internal {
      type internal;
      export next-hop-self;
      neighbor 10.0.0.3;
      neighbor 10.0.0.4;
      neighbor 10.0.0.6;
    }
    group toR1 {
      type external;
      import import-toR1;
      peer-as 65001;
      neighbor 10.1.12.1;
    }
  }
  isis {
    level 1 disable;
    interface all {
      level 2 metric 10;
    }
    interface lo0.0;
  }
}
policy-options {
  policy-statement next-hop-self {
    term change-next-hop {
      from neighbor 10.1.12.1;
      then {
        next-hop self;
      }
    }
  }
  policy-statement import-toR1 {
    term 1 {
      from {
        route-filter 100.100.1.0/24 exact;
      }
      then {
        local-preference 200;
      }
    }
  }
}

user@R4> show configuration
[...Output truncated...]
interfaces {
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.46.1/30;
      }
      family iso;
    }
  }
  so-0/0/2 {

```

```
        unit 0 {
            family inet {
                address 10.1.45.1/30;
            }
            family iso;
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.1.24.2/30;
            }
            family iso;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
            family iso {
                address 49.0001.1000.0000.0004.00;
            }
        }
    }
}
routing-options {
    [...Output truncated...]
    router-id 10.0.0.4;
    autonomous-system 65002;
}
protocols {
    bgp {
        group internal {
            type internal;
            local-address 10.0.0.4;
            export next-hop-self;
            neighbor 10.0.0.2;
            neighbor 10.0.0.3;
            neighbor 10.0.0.6;
        }
        group toR5 {
            type external;
            peer-as 65001;
            neighbor 10.1.45.2;
        }
    }
    isis {
        level 1 disable;
        interface all {
            level 2 metric 10;
        }
        interface lo0.0;
    }
}
policy-options {
    policy-statement next-hop-self {
        term change-next-hop {
            from neighbor 10.1.45.2;
            then {
                next-hop self;
            }
        }
    }
}
```

```

    }
  }
}

```

**Meaning** The sample output shows a basic BGP configuration on border routers **R2** and **R4**. Both routers have the AS (65002) included at the **[routing-options]** hierarchy level. Each router has two groups included at the **[protocols bgp group group]** hierarchy level. External peers are included in the external group, either **toR1** or **toR5**, depending on the router. Internal peers are included in the **internal** group. The underlying IGP protocol is IS-IS on both routers, and relevant interfaces are configured to run IS-IS.

Note that in the configuration on both routers, the router ID is manually configured to avoid duplicate router ID problems, and the **next-hop-self** statement is included to avoid any BGP next-hop reachability problems.



# Routing Protocol Process Memory FAQs

- [Routing Protocol Process Memory FAQs Overview on page 1107](#)
- [Routing Protocol Process Memory FAQs on page 1108](#)

## 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 1108](#)

## 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.

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 48 on page 1110](#) 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 48 on page 1110 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 48: show system processes extensive Output Fields

Field Name	Field Description
<b>Mem</b>	Information about physical and virtual memory allocation.
<b>Active</b>	Memory allocated and actively used by the program.
<b>Inact</b>	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.
<b>Wired</b>	Memory that is not eligible to be swapped, usually used for in-kernel memory structures and/or memory physically locked by a process.
<b>Cache</b>	Memory that is not associated with any program and does not need to be swapped before being reused.
<b>Buf</b>	Size of memory buffer used to hold data recently called from the disk.
<b>Free</b>	Memory that is not associated with any programs. Memory freed by a process can become <b>Inactive</b> , <b>Cache</b> , or <b>Free</b> , depending on the method used by the process to free the memory.
<b>Swap</b>	Information about swap memory. <ul style="list-style-type: none"> <li>• Total—Total memory available to be swapped to disk.</li> <li>• Used—Memory swapped to disk.</li> <li>• Free—Memory available for further swap.</li> </ul>

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 49 on page 1111](#) 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 49 on page 1111](#) describes the output fields for the **show task memory** command. Output fields are listed in the approximate order in which they appear.

**Table 49: show task memory Output Fields**

Field Name	Field Description
Memory Currently In Use	Memory currently in use. Dynamically allocated memory plus the <b>DATA</b> 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, inact 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 1107](#)

## PART 5

# Index

- [Index on page 1117](#)



# Index

## Symbols

#, comments in configuration statements.....	xxv
( ), in syntax descriptions.....	xxv
< >, in syntax descriptions.....	xxv
[ ], in configuration statements.....	xxv
{ }, in configuration statements.....	xxv
(pipe), in syntax descriptions.....	xxv

## A

accept	
firewall filters	
action.....	552
accept-remote-nexthop statement.....	604
usage guidelines.....	371
accepted-prefix-limit statement.....	605
usage guidelines.....	541
action modifiers, firewall filters.....	552
active	
BGP protocol session state.....	1092
active routes.....	8, 257
add-path statement	
BGP	
usage guidelines.....	393
administrative distance.....	250
BGP See preference statement	
advertise-external statement.....	608
usage guidelines.....	235, 304
advertise-from-main-vpn-tables statement.....	610
advertise-inactive statement.....	611
usage guidelines.....	234, 253, 298
advertise-peer-as statement.....	612
usage guidelines.....	200, 237, 312
advertisements, displaying .....	892
advertising multiple paths to a destination	
BGP.....	607, 729, 759
aggregate routes	
preferences.....	250
aggregate-label statement.....	613
AIGP	
BGP.....	152

aigp statement	
BGP.....	614, 616
usage guidelines.....	152
aigp-originate statement	
BGP	
usage guidelines.....	152
all (tracing flag).....	779
allow statement.....	619
always compare, BGP MED option.....	94
always-compare-med option.....	8, 257
apply-path statement	
firewall filter match condition.....	465
arithmetic and relational operators	
for monitor traffic command.....	812
AS path	
ignoring in route selection.....	260
AS paths	
matching regular expressions, displaying.....	899
as-override	
usage guidelines.....	191
as-override statement.....	620
as-path (tracing flag).....	776
as-path-ignore	
usage guidelines.....	8, 257, 260
as-path-ignore option.....	730
ASN	
BGP community routes, displaying.....	906
aspath, BGP protocol tracing flag.....	1094
ASs	
paths.....	6
operations, tracing.....	776
private, removing.....	268, 269, 750
ASs (autonomous systems)	
breaking into confederations.....	441
attribute set messages	
disabling.....	209
authentication See of routes	
algorithm	
BGP.....	450
BGP.....	7, 617, 621
keychains	
BGP.....	450
MD5	
BGP.....	450
authentication configuration	
BFD.....	360

authentication-algorithm statement		authentication keychain.....	625
BGP.....	623	autonomous system override.....	620
usage guidelines.....	450	best external route	
usage guidelines.....	577	advertising.....	235, 304
authentication-key statement		BFD.....	349, 350, 617, 621, 627, 676, 693, 761
BGP		BGP_L2VPN_AD_NLRI.....	626
usage guidelines.....	450	CLNS.....	569, 571
BGP and BMP.....	624	communities	
usage guidelines.....	578	policy, routing.....	223
authentication-key-chain		community ASN, displaying routes.....	906
usage guidelines.....	578	community name, displaying routes.....	908
authentication-key-chain statement.....	625	community remove.....	319
BGP		community-count.....	326
usage guidelines.....	450	configuration, verifying.....	1062, 1077
usage guidelines.....	450	damping parameters	
auto-discovery-only statement		clearing.....	799
BGP.....	626	displaying.....	873, 879
autonomous system number See ASN		damping routes, displaying.....	910
<b>B</b>		delete group internal neighbor	
bandwidth management		command.....	1070
BGP.....	339, 383	description.....	57
best routes, displaying.....	901	EBGP IPv6 peering.....	41
BFD		edit protocol bgp command.....	1070
authentication configuration.....	360	enabling on router.....	631
protocol.....	349, 350	error handling for update messages.....	527
sessions		external (EBGP).....	5
clearing.....	798	FEC 129.....	626
displaying.....	829	filtering paths to a destination.....	740
with IBGP.....	349, 350	graceful restart.....	654
bfd-liveness-detection statement		groups.....	33, 92, 655
BFD		general information, displaying.....	839
threshold.....	772	traffic statistics, displaying.....	846
transmit-interval.....	783	hold time.....	7, 663
BGP.....	627	identifier.....	7
usage guidelines.....	350	idle-after-switch-over statement.....	666
BGP See multipath See secure interdomain routing		ignoring the AS path attribute in route	
administrative distance.....	250	selection.....	260
advertise-peer-as.....	200, 312	indirect next hops.....	371
advertising multiple paths to a		injecting OSPF routes into BGP.....	220
destination.....	392, 393, 607, 729, 759	internal.....	56
aggregator path attribute.....	718	internal (IBGP).....	5
AIGP attribute.....	152	IP address.....	7
applying routing policies.....	211	IPsec.....	456, 672
AS numbers, peers.....	732	IPv6.....	543
ASs See ASs		logical systems.....	41
attribute set messages.....	209	keepalive messages.....	8, 776
authentication.....	7, 450, 624	LDP-based Layer 2 VPN and VPLS update	
authentication algorithm.....	623	messages.....	626
		local address.....	57, 680

- local AS.....131, 136, 146
- local interface.....686
- local preference.....79, 80
- loop detection.....209
- MED.....95, 107, 121
- messages.....7
- MP-BGP.....537, 647
- MTU discovery.....706
- multihop sessions.....708, 721
  - ttl.....785
- multipath configuration.....339, 365, 366, 383
- neighbors
  - clearing connections.....800
  - displaying.....848
- neighbors BGP, peers See BGP, peers
- NLRI.....8
  - IPv4 VPN.....537
  - IPv6 VPN.....537
- open messages.....7, 497, 728
- origin AS validation.....480
- outbound route filter.....237
- outbound route filters
  - interoperability.....632
- overview.....4
- packets, tracing.....776
- passive mode.....497
- path attributes.....6, 8
- peers.....5, 92, 713
- point-to-point peer session (configuration
  - editor).....34
- policy, routing.....646, 667
- precedence.....233
- preferences.....250, 252, 333, 736
- prefix-limit
  - accepted.....541
  - received.....541
- private AS.....268, 269
- receiving multiple paths to a destination.....746
- resolve routes to other tables.....542
- route reflection.....636, 719
- route target filtering.....757
- route validation
  - information,
    - displaying.....1040, 1042, 1044, 1046, 1049
- routes.....6
- routing tables
  - delays in exchanging routes.....236, 725
  - nonactive routes.....234, 253, 298, 611
  - retaining routes.....674
- routing to CE devices.....191
- scaling.....735
- secure interdomain routing
  - group.....658
  - hold-time period.....665
  - local address.....683
  - maximum prefix length.....697
  - maximum sessions.....696
  - origin AS.....724
  - port.....733
  - preference.....737
  - record.....747
  - record lifetime.....748
  - refresh time.....749
  - session.....760
  - static records.....763
  - tracing.....651, 652, 781
  - validation.....788
  - validation state.....789
- session drops.....497
- sessions, checking .....1061, 1072
- set local AS number.....684
- set local-address command.....1070
- show bgp summary command.....1061, 1072
- show configuration command.....1062, 1077
- show route detail command.....1068
- show route receive protocol bgp
  - command.....1069, 1095
- summary information, displaying.....867
- supported software standards.....13
- system log messages.....497
- table
  - clearing.....802
- TCP.....4
- TCP segment size.....468
- traceroute command.....1071, 1098
- tracing operations.....776
  - 4-byte AS events.....583
  - BFD protocol events.....583
  - damping operations.....583
  - description.....583
  - graceful restart.....583
  - keepalive messages.....583
  - NSR synchronization.....583
  - open PDUs.....583
  - policy processing.....583
  - protocol task processing.....583
  - protocol timer processing.....583
  - refresh PDUs.....583

route information.....	583	network	
state transitions.....	583	configuration topology, figure .....	1076
update PDUs.....	583	topology, figure .....	1074, 1085
traffic, verifying .....	1071, 1097	open statement.....	1083
type, group.....	787	protocol statement.....	1083
update messages.....	8	run show log command.....	1084
version supported.....	4	send statement.....	1082
VPNs		session	
preventing session flaps.....	497	problems.....	1083, 1092
with BFD.....	349, 350	states, table .....	1092
BGP (Border Gateway Protocol)		set flag command.....	1082
confederations See BGP confederations		show configuration command	
for CLNS VPN NLRI.....	569	border router.....	1102
internal peer session (configuration		internal router.....	1100
editor).....	57	show route advertising-protocol bgp	
peering sessions See BGP peers; BGP sessions		command.....	1096
point-to-point internal peer session		show route command	
logical systems.....	68	EBGP over IBGP.....	1088, 1098
requirements.....	12	IGP cost.....	1089
route reflectors See BGP route reflectors		local preference.....	1086
route-flap damping.....	505	MED.....	1087
BGP confederations		show route forwarding-table command.....	1090
creating (configuration editor).....	442	show route receive-protocol bgp	
description.....	441	command.....	1096
route-flap damping.....	505	state transitions, logging.....	1091
BGP groups		traceoptions statement.....	1082, 1083, 1093
confederations (configuration editor).....	442	tracing	
BGP layer		configuring.....	1093
broken network topology, figure .....	1060	flags, table.....	1095
BGP Monitoring Protocol.....	634	update statement.....	1082
configuring.....	580	BGP route reflectors	
displaying		cluster of clusters.....	420
statistics.....	837	creating (configuration editor).....	421, 437
BGP monitoring protocol version 3 See BMP version		description.....	419
3		multiple clusters.....	420
BGP peers		BGP sessions	
external (configuration editor).....	34	internal.....	68
internal.....	68	internal (configuration editor).....	57
internal (configuration editor).....	57	point-to-point external (configuration	
point-to-point connections.....	33	editor).....	34
BGP protocol		sample peering session.....	33
checklist for verifying.....	1057	bgp statement.....	631
detail statement.....	1083	bgp-error-tolerance statement.....	527
edit protocol bgp traceoptions		bgp-orf-cisco-mode	
command.....	1082	usage guidelines.....	237
edit protocol command.....	1083	bgp-orf-cisco-mode statement.....	632
establishment issues .....	1083	BGP_L2VPN_AD_NLRI.....	626
flag statement.....	1083	Bidirectional Forwarding Detection See BFD	

- 
- BMP**
- authentication algorithm.....623
  - authentication keychain.....625
  - tracing operations.....779
- bmp statement**.....634
- usage guidelines.....580
  - usage guidelines.....577
- BMP version 3**
- configuration.....577
- Border Gateway Protocol** *See* BGP
- braces, in configuration statements**.....xxv
- brackets**
- angle, in syntax descriptions.....xxv
  - square, in configuration statements.....xxv
- C**
- checklist for**
- BGP protocol, verifying.....1057
- Cisco non-deterministic, BGP MED option**.....94
- cisco-non-deterministic option**.....8, 257, 730
- clear bfd adaptation command**.....797
- clear bfd session command**.....798
- clear bgp damping command**.....799
- clear bgp neighbor command**.....800
- clear bgp table command**.....802
- clear validation database command**.....804
- clear validation session command**.....805
- clear validation statistics command**.....806
- CLNS**.....673
- BGP.....569, 571
- CLNS (Connectionless Network Service) VPNs**
- BGP, to carry CLNS VPN NLRI.....569
- cluster statement**.....636
- usage guidelines.....421, 437
- clusters** *See* BGP route reflectors
- commands for**
- BGP protocol, verifying.....1057
- comments, in configuration statements**.....xxv
- communities**
- policy, routing.....223
- community**
- bandwidth.....339, 383
  - example.....225
- community ASN, displaying routes**.....906
- community name, displaying routes**.....908
- community remove**.....319
- community-count match condition**.....326
- conditions**
- routing policy.....877
- confederation statement**
- usage guidelines.....442
- confederations** *See* BGP confederations
- connect, BGP protocol session state**.....1092
- connection-mode statement**.....637
- usage guidelines.....578
- conventions**
- text and syntax.....xxiv
- count (firewall filter action)**.....552
- curly braces, in configuration statements**.....xxv
- customer support**.....xxvi
- contacting JTAC.....xxvi
- D**
- damping**.....638, 776
- damping (tracing flag)**.....776
- damping parameters, BGP**
- clearing.....799
  - displaying.....873, 879
- damping routes, BGP**
- displaying.....910
- damping statement**.....638
- usage guidelines.....506, 515
- damping, BGP protocol tracing flag**.....1094
- default routing policy** *See* PTX Series
- delay-med-update statement**
- usage guidelines.....121
- delete group internal neighbor command**.....1070
- description statement**.....640
- usage guidelines.....57
- destination-port (firewall filter match condition)**.....550
- detail statement**
- BGP protocol.....1083, 1093
- detection-time statement**
- BGP.....627, 641
- disable statement**.....776
- BGP.....642
- discard (firewall filter action)**.....552
- documentation**
- comments on.....xxv
- down**.....779
- dscp (firewall filter match condition)**.....550
- E**
- EBGP** *See* BGP
- EBGP (external BGP)**
- route-flap damping.....505
- EBGP IPv6 peering, BGP**.....41

edit protocol bgp command .....	1083, 1091	flow statement.....	647
edit protocol bgp traceoptions command .....	1082, 1093	usage guidelines.....	554
edit protocols bgp command .....	1070	font conventions.....	xxiv
error (tracing flag).....	779	forwarding table	
error handling		route entries, displaying.....	956
for BGP update messages.....	527	forwarding-class (firewall filter action).....	552
established, BGP protocol session state.....	1093	forwarding-table statement	
establishment issues		usage guidelines.....	339, 366, 383
BGP protocol.....	1083	fragment-offset (firewall filter match condition).....	550
events		fragmentation	
BGP protocol state transition.....	1091	avoiding.....	468
except (firewall filter match condition).....	550	full mesh requirement	
explicit-null statement.....	644	fulfilling with confederations.....	441
export route information, displaying.....	934	fulfilling with route reflectors.....	419
export statement		<b>G</b>	
BGP.....	646	general (tracing flag)	
usage guidelines.....	234	BMP.....	779
forwarding table		generated routes	
usage guidelines.....	339, 366, 383	preferences.....	250
IS-IS		graceful-restart statement.....	654
usage guidelines.....	225	BGP.....	654
export statement, for routing policies.....	220	group statement	
external-router-id option.....	8, 257	BGP.....	655
<b>F</b>		Origin AS Validation.....	658
family statement		groups	
BGP.....	647	BGP	
usage guidelines.....	537	general information, displaying.....	839
FAQs		traffic statistics, displaying.....	846
routing protocol process memory.....	1107, 1108	<b>H</b>	
fault tolerance		hidden routes.....	1056
advertising multiple paths to a destination.....	392, 393	hidden routes, displaying.....	970
FEC 129.....	626	hold-down statement.....	659
file command		usage guidelines.....	578
logical systems.....	584	hold-down-interval	
file statement		BGP.....	661
Origin AS Validation.....	651	hold-time statement	
filtering paths to a destination		BGP.....	663
BGP.....	740	Origin AS Validation.....	665
flag statement		<b>I</b>	
Origin AS Validation.....	652	IBGP See BGP	
flag statements		overview.....	56
BGP protocol.....	1082, 1083	IBGP (internal BGP)	
flap damping.....	505	full mesh (configuration editor).....	33
parameters.....	506	icmp-code (firewall filter match condition).....	550
flow routes.....	550	icmp-type (firewall filter match condition).....	550
BGP.....	554		

- identifiers
    - BGP See BGP, identifier
  - idle, BGP protocol state.....1092
  - idle-after-switch-over statement.....666
  - IGP plus MED, BGP option.....94
  - import statement
    - BGP.....667
    - usage guidelines.....234
  - import statement, for routing policies.....220
  - include-mp-next-hop statement.....669
  - independent domain
    - attribute set messages.....209
  - inet-mdt statement
    - BGP address family.....670
  - ingress router
    - show bgp summary command.....1062, 1073
  - initiation-message statement.....671
    - usage guidelines.....578
  - install-to-fib
    - usage guidelines.....276
  - interface-group (firewall filter match condition).....550
  - ipsec-sa statement
    - BGP.....672
    - usage guidelines.....456
  - IPv6
    - BGP.....543
    - EBGP link-local peering.....41
    - logical systems.....41
  - IS-IS
    - authentication, displaying.....1037
    - policy, routing.....225
    - preferences.....225, 250
  - iso-vpn statement.....673
    - usage guidelines.....569, 571
- K**
- keep statement.....674
    - usage guidelines.....236
  - keepalive (tracing flag)
    - BGP.....776
  - keepalive messages.....8
  - keepalive, BGP protocol tracing flag.....1094
  - key-chain-name
    - BGP.....676
  - keychain
    - BGP.....450
    - overview.....449
- L**
- labeled-unicast statement.....678
  - layered model
    - BGP layer, figure .....1059
  - LDP
    - authentication algorithm.....623
  - LDP-based Layer 2 VPN and VPLS update
    - messages
      - BGP.....626
  - load balance
    - asymmetric.....339, 383
    - unequal.....339, 383
  - load balancing
    - advertising multiple paths to a destination.....392, 393
    - per-prefix.....276
  - load-balance statement
    - usage guidelines.....339, 366, 383
  - local AS
    - BGP.....131, 136, 146
  - local-address statement
    - BGP.....680
    - BMP.....682
    - Origin AS Validation.....683
    - usage guidelines.....57, 579
  - local-as statement.....684
    - usage guidelines.....136, 146
  - local-interface statement
    - BGP.....686
    - usage guidelines.....41
  - local-port statement
    - BMP.....687
    - usage guidelines.....579
  - local-preference statement.....688
    - usage guidelines.....80
  - log (firewall filter action).....552
  - log-updown statement.....689
    - BGP
      - usage guidelines.....497
  - logical operators
    - for monitor traffic command.....810
  - logical systems
    - EBGP
      - with IPv6 interfaces.....41
    - internal BGP.....68
    - viewing system files on.....584
  - logical-systems statement.....690
  - loop detection
    - BGP.....209

loops statement		
BGP address family.....	691	
loose-check		
BGP.....	693	
loss-priority (firewall filter action).....	552	
<b>M</b>		
malformed BGP attributes.....	527	
malformed-route-limitstatement		
for BGP.....	527	
malformed-update-log-interval statement		
for BGP.....	527	
manuals		
comments on.....	xxv	
match conditions		
firewall filters		
overview.....	550	
for monitor traffic command.....	809	
max-sessions statement		
Origin AS Validation.....	696	
maximum-length statement		
Origin AS Validation.....	697	
MBGP MVPNs.....	515	
MD5 authentication.....	450	
BGP.....	450	
MED See BGP		
MED (multiple exit discriminator)		
always compare option.....	94	
Cisco non-deterministic option.....	94	
plus IGP option.....	94	
med-igp-update-interval statement		
usage guidelines.....	121	
med-plus-igp statement.....	730	
usage guidelines.....	8, 257	
members statement		
usage guidelines.....	442	
metric statement		
BGP		
usage guidelines.....	107	
metric-out statement		
BGP.....	698	
usage guidelines.....	95	
minimum-interval.....	700	
BFD, transmit-interval.....	702	
usage guidelines.....	350	
minimum-interval statement		
BGP.....	627	
minimum-receive-interval		
BFD.....	704	
minimum-receive-interval statement		
BFD (BGP)		
usage guidelines.....	350	
BGP.....	627	
monitor statement		
BMP.....	705	
usage guidelines.....	579	
monitor traffic command.....	807	
MP-BGP.....	537, 647	
MPLS		
ultimate-hop popping.....	644	
mtu-discovery statement.....	706	
multihop		
BGP.....	241	
multihop statement.....	708, 721	
usage guidelines.....	241	
multipath statement.....	710	
usage guidelines.....	339, 366, 383	
multiplier		
BFD.....	711	
multiplier statement		
BFD (BGP)		
usage guidelines.....	350	
BGP.....	627	
multiprotocol BGP		
IPv6 example.....	543	
multiprotocol BGP (MP-BGP).....	537, 647	
<b>N</b>		
neighbor statement		
BGP.....	713	
neighbors		
BGP.....	5	
network layer reachability information See BGP,		
NLRI See NLRI		
network topology		
BGP protocol, figure.....	1074, 1084	
networks		
sample BGP confederations.....	442	
sample BGP MED use.....	93	
sample BGP peer session.....	33	
sample BGP route reflector (one cluster).....	420	
sample BGP route reflectors (cluster of		
clusters).....	421	
sample BGP route reflectors (multiple		
clusters).....	420	
next hops		
routes sent to, displaying.....	986	

- NLRI
  - BGP\_L2VPN\_AD\_NLRI.....626
- NLRI (network layer reachability information), BGP
  - for CLNS.....569
- NLRI, BGP.....8
- nlri-route-type statement
  - usage guidelines.....515
- no-adaptation
  - BFD.....716
  - BFD (BGP)
    - usage guidelines.....350
- no-adaptation statement
  - BGP.....627
- no-advertise-peer-as statement.....717
  - usage guidelines.....200, 237, 312
- no-aggregator-id statement.....718
- no-client-reflect statement.....719
- no-install-to-fib
  - usage guidelines.....276
- no-malformed-route-limit statement
  - for BGP.....527
- no-prepend-global-as statement
  - usage guidelines.....136
- no-validate statement.....723
- normal (tracing flag)
  - BMP.....779
- O**
  - open messages, BGP.....7
  - open statement, BGP protocol .....1083
  - open, BGP protocol tracing flag.....1094
  - openConfirm, BGP session state.....1093
  - openSent, BGP protocol session state.....1092
- ORF
  - BGP.....237
- origin AS validation
  - BGP.....480
- Origin AS Validation
  - group.....658
  - hold-time period.....665
  - local address.....683
  - maximum prefix length.....697
  - maximum sessions.....696
  - origin AS.....724
  - port.....733
  - preference.....737
  - record.....747
  - record lifetime.....748
  - refresh time.....749
  - session.....760
  - static records.....763
  - tracing.....651, 652, 781
  - validation.....788
  - validation state.....789
- origin validation
  - RPKI.....804, 805, 806, 817
- origin-autonomous-system statement
  - Origin AS Validation.....724
- OSPF
  - preferences.....250
- OSPF (Open Shortest Path First)
  - injecting OSPF routes into BGP.....220
- out-delay statement.....725
  - usage guidelines.....236
- outbound-route-filter
  - usage guidelines.....237
- outbound-route-filter statement
  - BGP.....727
- P**
  - packet headers, transmitted, displaying.....807
  - packet-length (firewall filter match
    - condition).....550
  - packets
    - BGP protocol tracing flag.....1095
    - received .....1082
    - sent.....1082
  - packets (tracing flag)
    - BGP.....776
    - BMP.....779
  - parentheses, in syntax descriptions.....xxv
  - passive statement
    - BGP.....728
    - usage guidelines.....497
  - path attributes, BGP.....6, 8
  - path-count statement.....729
    - BGP
      - usage guidelines.....393
  - path-selection statement.....730
    - usage guidelines.....8, 257
  - peer-as statement.....732
  - peering sessions See BGP peers; BGP sessions
  - per-packet load balancing.....339, 366, 383
  - per-packet statement
    - usage guidelines.....339, 366, 383
  - per-prefix load balancing.....276
  - policers
    - firewall filter action.....552

policy (tracing flag)		
BMP.....	779	
policy, routing		
BGP.....	646, 667	
communities.....	223	
IS-IS.....	225	
precedence.....	233	
port (firewall filter match condition).....	550	
port statement		
Origin AS Validation.....	733	
post-policy statement.....	742	
usage guidelines.....	579	
pre-policy statement.....	734	
usage guidelines.....	579	
precedence (firewall filter match condition).....	550	
precision-timers statement		
BGP.....	735	
preference statement		
BGP.....	736	
usage guidelines.....	252, 333	
Origin AS Validation.....	737	
preferences		
active routes.....	8, 257	
aggregate routes		
generated routes.....	250	
default.....	250	
IS-IS.....	250	
modifying		
with configuration statements.....	250	
RIP.....	250	
static routes.....	250	
prefix list statement		
firewall filter match condition.....	465	
prefix-based		
usage guidelines.....	237	
prefix-limit statement.....	738	
usage guidelines.....	541, 549	
prefix-policy statement.....	740	
BGP		
usage guidelines.....	393	
priority statement		
BMP.....	741	
usage guidelines.....	579	
private statement		
usage guidelines.....	146	
processes		
restarting.....	818	
propagation, suppressing.....	505	
protocol bgp statement.....	1083	
protocols		
firewall filter match condition.....	550	
match condition		
firewall filters.....	550	
protocols statement.....	744	
PTX Series		
default routing policy.....	276	
<b>R</b>		
real-time monitoring		
traffic.....	807	
receive statement.....	746	
BGP		
usage guidelines.....	393	
receive statements		
BGP protocol statement.....	1082	
receiving multiple paths to a destination		
BGP.....	746	
record statement		
Origin AS Validation.....	747	
record-lifetime statement		
Origin AS Validation.....	748	
redirected routes.....	250	
refresh-time statement		
Origin AS Validation.....	749	
regular expressions		
AS paths, displaying matching routes.....	899	
reject		
firewall filters		
action.....	552	
remove-private statement.....	750	
usage guidelines.....	269	
replication		
of BGP configuration.....	862	
request validation policy command.....	817	
resolve-vpn statement.....	752	
usage guidelines.....	542	
resource public key infrastructure See RPKI		
restart command.....	818	
restart-time statement.....	753	
restarting		
software processes.....	818	
rib statement		
BGP.....	754	
rib-group statement		
BGP.....	755	
usage guidelines.....	537	
RID.....	1067, 1081	

- 
- RIP
    - preferences.....250
  - route (tracing flag)
    - BMP.....779
  - route advertisements, displaying.....892
  - route authentication
    - peering sessions.....449
  - route injection.....220
  - route redistribution.....220
  - route reflector
    - in a Layer 3 VPN.....191
  - route reflectors *See* BGP route reflectors
    - BGP.....421, 437
  - route resolution
    - BGP.....752
  - route, displaying
    - next-hop.....986
  - route-flap damping.....505
    - parameters.....506
  - route-monitoring (tracing flag)
    - BMP.....779
  - route-monitoring statement.....756
    - usage guidelines.....579
  - route-target statement.....757
  - routers
    - border router BGP, configuring.....1102
    - configuration
      - BGP protocol.....1075
    - internal router BGP, configuring.....1099
  - routes, displaying
    - active.....881
    - active path.....887
    - advertising protocol.....892
    - all.....897
    - AS paths
      - regular expressions, matching.....899
    - best.....901
    - brief information.....904
    - community ASN.....906
    - community name.....908
    - damping, BGP.....910
    - detailed information.....915
    - extensive information.....937
    - flow validation.....954
    - hidden.....970
    - in a specific routing table.....1020
    - in the forwarding table.....956
    - inactive path.....973
    - inactive prefix.....976
    - instances.....978
    - learned from a specific protocol.....1000
    - matching the specified address.....932
    - not associated with a community.....992
    - policy-based route export.....934
    - received through a neighbor.....1012
    - sent to a specific interface.....995
    - terse information.....1034
  - routes, hidden
    - understanding.....1056
  - routing policies
    - applying.....220
    - configuration tasks.....220, 506
    - displaying.....875
    - export statement.....220
    - import statement.....220
    - injecting routes from one protocol into
      - another.....220
    - reducing update messages with flap
      - damping.....505
    - route redistribution.....220
    - route-flap damping.....505
    - testing the configuration for.....1051
  - routing policy
    - applying to BGP.....211
  - routing protocol process memory
    - FAQ.....1107, 1108
  - routing solutions
    - BGP confederations, for scaling
      - problems.....442
    - BGP route reflectors, for scaling
      - problems.....421, 437
    - reducing update messages with flap
      - damping.....505
  - routing tables
    - BGP, delays in exchanging routes.....236
    - group.....755
    - nonactive routes, exchanging with
      - BGP.....234, 253, 298, 611
  - routing-instance (firewall filter action).....552
  - routing-instances statement.....758
  - RPKI
    - information,
      - displaying.....1040, 1042, 1044, 1046, 1049
  - RSVP
    - preferences.....250
  - run show log command
    - BGP protocol.....1083, 1084, 1094

## S

sample (firewall filter action).....	552
secure interdomain routing See BGP	
send statement.....	607, 759
BGP	
usage guidelines.....	393
BGP protocol.....	1082
session statement	
Origin AS Validation.....	760
session states, BGP protocol .....	1092
session-mode statement	
BGP.....	761
sessions, checking BGP.....	1061, 1072
set flag command	
BGP protocol .....	1082
set flag update detail command.....	1093
set flag update send command .....	1082
set local-address command .....	1070
set log-updown command.....	1091
set traceoptions flag open detail command.....	1083
show bfd session command.....	829
show bgp bmp command.....	837
show bgp group command.....	839
show bgp group traffic-statistics command.....	846
show bgp neighbor command.....	445, 848
explanation.....	446
show bgp replication command.....	862
show bgp summary	
command.....	447, 867, 1061, 1072
explanation.....	447
show configuration command .....	1062, 1077
BGP protocol	
border router.....	1102
internal router.....	1100
show policy command.....	875
show policy conditions command.....	877
show policy damping command.....	873, 879
show route active-path command.....	887
show route advertising-protocol bgp	
command.....	1096
show route advertising-protocol command.....	892
show route all command.....	897
show route aspath-regex command.....	899
show route best command.....	901
show route brief command.....	904
show route command.....	881
BGP protocol	
EBGP over IBGP.....	1088, 1098
IGP cost.....	1089
local preference.....	1086
MED.....	1087
show route community command.....	906
show route community-name command.....	908
show route damping command.....	910
show route detail command.....	915, 1068
show route exact command.....	932
show route export command.....	934
show route extensive command.....	937
show route flow validation command.....	954
show route forwarding-table	
command.....	956, 1090
show route hidden command.....	970
show route inactive-path command.....	973
show route inactive-prefix command.....	976
show route instance command.....	978
show route next-hop command.....	986
show route no-community command.....	992
show route output command.....	995
show route protocol command.....	1000
show route receive protocol bgp	
command.....	1069, 1095
show route receive-protocol bgp command.....	1096
show route receive-protocol command.....	1012
show route table command.....	1020
show route terse command.....	1034
show security keychain command.....	1037
show validation database command.....	1040
show validation group command.....	1042
show validation replication database	
command.....	1044
show validation session command.....	1046
show validation statistics command.....	1049
source-port (firewall filter match condition).....	550
stack bit value.....	1072, 1098
stale-routes-time statement.....	762
state	
BMP.....	779
state transition events .....	1091
stateless firewall filters	
accepting Routing Engine traffic from trusted	
sources	
example: blocking TCP access.....	460
example: blocking Telnet and SSH	
access.....	465
examples	
blocking TCP access.....	460
blocking Telnet and SSH access.....	465
states, BGP protocol.....	1092

- static routes
    - preferences.....250
  - static statement
    - Origin AS Validation.....763
  - station statement.....764
    - usage guidelines.....577
  - station-address statement.....765
    - usage guidelines.....577
  - station-port statement.....766
    - usage guidelines.....578
  - statistics (tracing flag)
    - BMP.....779
  - statistics-timeout statement.....767
    - usage guidelines.....580
  - sub-ASs, BGP.....441
  - subautonomous systems, BGP.....441
  - support, technical See technical support
  - syntax conventions.....xxiv
  - syslog (firewall filter action).....552
- T**
- task (tracing flag)
    - BMP.....779
  - tcp-mss statement.....769
    - BGP
      - usage guidelines.....468
  - technical support
    - contacting JTAC.....xxvi
  - test policy command.....1051
  - threshold
    - BFD.....772
    - BGP.....770
  - threshold statement
    - BFD (BGP)
      - usage guidelines.....350
    - BGP.....627
  - time-to-live.....1072, 1098
  - timer (tracing flag)
    - BMP.....779
  - topology
    - sample BGP confederations.....442
    - sample BGP MED use.....93
    - sample BGP peer session.....33
    - sample BGP route reflector (one cluster).....420
    - sample BGP route reflectors (cluster of clusters).....421
    - sample BGP route reflectors (multiple clusters).....420
  - topology statement
    - Multitopology Routing
      - OSPF.....774
  - trace files
    - logical systems
      - .....584
  - traceoptions statement
    - BGP.....776
      - description.....583
    - BGP protocol.....1082, 1083
    - BMP.....779
    - Origin AS Validation.....781
    - usage guidelines.....589
  - traceroute command .....1071, 1098
  - tracing flags
    - all.....779
    - as-path.....776
    - BGP protocol, table .....1095
    - damping.....776
    - down.....779
    - error.....779
    - event.....779
    - general.....779
    - keepalive
      - BGP.....776
    - normal.....779
    - packets.....779
      - BGP.....776
    - policy.....779
    - route.....779
    - route-monitoring.....779
    - state.....779
    - statistics.....779
    - task.....779
    - timer.....779
    - up.....779
    - write.....779
  - tracing operations
    - BGP.....583, 776
    - BMP.....779
  - traffic, real-time monitoring.....807
  - traffic, verifying BGP .....1071, 1097
  - traffic-statistics statement.....782
  - transmit-interval
    - BFD.....783
  - transmit-interval statement
    - BGP.....627
  - ttl statement.....785
  - type statement.....787

**U**

unequal load balancing.....	339, 383
up (tracing flag)	
BMP .....	779
update messages	
BGP .....	8
update statement, BGP protocol .....	1082, 1093
update, BGP protocol tracing flag.....	1095

**V**

validation statement	
Origin AS Validation.....	788
usage guidelines.....	480, 554
validation-state statement	
Origin AS Validation.....	789
verification	
BGP session flap prevention.....	504, 532
BMP .....	582
IS-IS policy.....	217
network interfaces.....	184, 413
tracing.....	589
version	
BFD.....	790
version statement	
BFD (BGP)	
usage guidelines.....	350
BGP .....	627
vpn-apply-export statement.....	791
VPNs	
BGP	
preventing session flaps.....	497
VRF export policy.....	791

**W**

write (tracing flag)	
BMP .....	779