

AOS-CX 10.06 IP Routing Guide

6300, 6400, 8320, 8325, 8360 Switch Series



a Hewlett Packard
Enterprise company

Part Number: 5200-7701
Published: November 2020
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Optane™, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

All third-party marks are property of their respective owners.

Chapter 1 About this document	14
Applicable products	14
Latest version available online	14
Command syntax notation conventions	14
About the examples	15
Identifying switch ports and interfaces	16
Identifying modular switch components	16
 Chapter 2 Virtual Routing and Forwarding (VRF)	 18
Overview	18
Adding or deleting a VRF	18
IPv4 static route addition or deletion in a VRF	18
IPv6 static route addition or deletion in a VRF	19
Attaching or detaching a port from a VRF	20
Viewing VRF information	20
An example of the VRF information provided by the show running-config command	21
VRF commands	21
ip route vrf	21
ipv6 route vrf	22
show ip route	23
show ipv6 route	25
show vrf	26
vrf	27
vrf attach	28
 Chapter 3 Loopback	 30
Overview	30
Loopback commands	30
interface loopback	30
ip address	31
ipv6 address	31
vrf attach <VRF-NAME>	32
show interface loopback	32
 Chapter 4 Static routing	 34
Overview	34
Default route	34
Recursive static routes	34
Configuration concepts	34
Configuration example procedure	35
Configuration prerequisites	35
Basic static route configuration example	35
Static routing commands	36
ip route	36
ip route bfd	37
ip route distance	38

ip route tag.....	39
ip route vrf.....	39
ipv6 route.....	40
ipv6 route distance.....	42
ipv6 route tag.....	42
ipv6 route vrf.....	43
show ip route.....	44
show ipv6 route.....	46

Chapter 5 Open Shortest Path First version 2 (OSPFv2)..... 48

Overview.....	48
How OSPFv2 protocol works.....	49
OSPFv2 concepts.....	49
OSPFv2 Link-state advertisement (LSA) types.....	49
OSPFv2 router types.....	50
OSPFv2 area types.....	50
OSPFv2 configuration task list.....	51
Configuring OSPF on the routing switch.....	52
Assigning the routing switch to an OSPF area.....	53
Setting OSPF network for the area.....	53
Creating an OSPF virtual link for an area.....	54
Configuring external route redistribution and control.....	54
Configuring area ranges on an ABR to reduce advertisements to the backbone.....	55
Influencing route choice by changing the administrative distance.....	55
Configuring graceful restart of OSPF routing.....	56
Configuring OSPF interface settings.....	56
Configuring OSPF interface authentication.....	57
Configuring OSPF virtual link settings.....	58
Configuring OSPF authentication on a virtual link.....	58
Configuring all OSPF interfaces as passive.....	59
Configuring SPF throttling timers.....	59
Viewing OSPFv2 information.....	60
Clearing OSPF statistics on a switch.....	61
An example of the OSPFv2 information in the show running-config command.....	61
OSPFv2 commands.....	62
area.....	62
area default-metric.....	63
area nssa.....	64
area range.....	65
area stub.....	66
area virtual-link.....	67
authentication.....	67
authentication-key.....	68
clear ip ospf neighbors	69
clear ip ospf statistics	70
dead-interval.....	71
default-information originate.....	71
default-information originate always.....	72
default-metric.....	72
disable.....	73
distance.....	74
enable.....	74
graceful-restart.....	75
hello-interval.....	75

ip ospf area.....	76
ip ospf authentication.....	77
ip ospf authentication-key.....	78
ip ospf cost.....	79
ip ospf dead-interval.....	79
ip ospf hello-interval.....	80
ip ospf keychain.....	81
ip ospf message-digest-key md5.....	81
ip ospf network.....	82
ip ospf passive.....	83
ip ospf priority.....	84
ip ospf retransmit-interval.....	84
ip ospf shutdown.....	85
ip ospf transit-delay.....	86
keychain.....	86
max-metric router-lsa.....	87
maximum-paths.....	88
message-digest-key md5.....	88
passive-interface default.....	89
redistribute.....	90
reference-bandwidth.....	91
retransmit-interval.....	92
rfc1583-compatibility.....	92
router ospf.....	93
router-id.....	94
show ip ospf.....	94
show ip ospf border-routers.....	95
show ip ospf interface.....	96
show ip ospf lsdb.....	98
show ip ospf neighbors.....	102
show ip ospf routes.....	104
show ip ospf statistics.....	106
show ip ospf statistics interface.....	107
show ip ospf virtual-links.....	108
summary-address.....	109
timers lsa-arrival.....	110
timers throttle lsa.....	111
timers throttle spf.....	112
transit-delay.....	113
trap-enable.....	113

Chapter 6 Open Shortest Path First version 3 (OSPFv3)..... 115

Overview.....	115
How OSPFv3 protocol works.....	116
OSPFv3 concepts.....	116
OSPFv3 Link-state advertisement (LSA) types.....	116
OSPFv3 area types.....	118
OSPFv3 router types.....	119
OSPFv3 configuration task list.....	120
Configuring OSPFv3 on the routing switch.....	120
Creating an OSPFv3 area.....	121
Setting OSPFv3 network for the area.....	121
Configuring external route redistribution and control.....	122
Configuring area ranges on an ABR to reduce advertisements to the backbone.....	123

Influencing route choice by changing the administrative distance.....	123
Configuring graceful restart.....	124
Configuring OSPFv3 virtual link settings.....	124
Configuring OSPFv3 interface settings.....	125
Configuring BFD for OSPFv3.....	126
Examples.....	126
Configuring all OSPFv3 interfaces as passive.....	127
Configuring SPF throttling timers.....	127
Viewing OSPFv3 information.....	127
Clearing OSPFv3 statistics on a switch.....	128
OSPFv3 commands.....	128
area.....	128
area authentication ipsec.....	129
area encryption ipsec.....	130
area nssa.....	131
area range.....	132
area stub.....	133
area virtual-link.....	134
area default-metric.....	135
authentication ipsec spi.....	136
clear ipv6 ospfv3 neighbors	136
clear ipv6 ospfv3 statistics	137
dead-interval.....	138
default-metric.....	139
disable.....	139
distance.....	140
enable.....	140
encryption ipsec spi.....	141
default-information originate.....	142
default-information originate always.....	142
graceful-restart.....	143
hello-interval.....	144
ipv6 ospfv3 area.....	145
ipv6 ospfv3 authentication null.....	145
ipv6 ospfv3 authentication ipsec.....	146
ipv6 ospfv3 cost.....	147
ipv6 ospfv3 dead-interval.....	147
ipv6 ospfv3 encryption ipsec.....	148
ipv6 ospfv3 encryption null.....	149
ipv6 ospfv3 hello-interval.....	150
ipv6 ospfv3 network.....	150
ipv6 ospfv3 passive.....	151
ipv6 ospfv3 priority.....	152
ipv6 ospfv3 retransmit-interval.....	152
ipv6 ospfv3 shutdown.....	153
ipv6 ospfv3 transit-delay.....	154
maximum-paths.....	154
max-metric router-lsa.....	155
passive-interface default.....	156
redistribute.....	156
reference-bandwidth.....	157
retransmit-interval.....	158
router-id.....	159
router ospfv3.....	159
show ipv6 ospfv3.....	160

show ipv6 ospfv3 border-routers.....	161
show ipv6 ospfv3 interface.....	162
show ipv6 ospfv3 neighbors.....	164
show ipv6 ospfv3 routes.....	166
show ipv6 ospfv3 statistics.....	167
show ipv6 ospfv3 statistics interface.....	168
show ipv6 ospfv3 virtual-links.....	170
summary-address.....	171
timers lsa-arrival.....	172
timers throttle lsa.....	173
timers throttle spf.....	173
transit-delay.....	175
trap-enable.....	175

Chapter 7 Border Gateway Protocol (BGP)..... 177

Overview.....	177
Autonomous system numbers.....	177
BGP sessions.....	177
Inter-router communication.....	178
BGP messages.....	178
BGP neighbor states.....	178
Injecting routes/prefixes into the BGP table.....	179
Path attributes.....	179
BGP best-path calculation.....	181
Loop prevention.....	181
Route policies.....	181
Resetting BGP sessions.....	181
IBGP full mesh requirement.....	182
Route reflectors.....	183
Loop prevention in route reflectors.....	183
BGP peer groups.....	184
BGP communities.....	184
Aggregate routes.....	184
BGP Graceful-Restart and high availability.....	184
Basic BGP configuration.....	185
Address families.....	185
Scale limits.....	185
BGP commands.....	186
address-family.....	186
aggregate-address.....	187
bgp always-compare-med.....	188
bgp asnotation dotted.....	189
bgp asnotation dotted-plus.....	189
bgp bestpath as-path ignore.....	190
bgp bestpath as-path multipath-relax.....	190
bgp bestpath compare-routerid.....	191
bgp bestpath med confed.....	192
bgp bestpath med missing-as-worst.....	192
bgp cluster id.....	193
bgp confederation.....	194
bgp confederation peers.....	194
bgp dampening.....	195
bgp default local-preference.....	196
bgp deterministic-med.....	197

bgp fast-external-fallover.....	197
bgp graceful-restart restart-time.....	198
bgp graceful-restart stalepath-time.....	198
bgp log-neighbor-changes.....	199
bgp maxas-limit.....	200
bgp router-id.....	200
clear bgp.....	201
disable enable.....	202
distance bgp.....	202
maximum-paths.....	203
neighbor activate.....	204
neighbor advertisement-interval.....	204
neighbor add-paths.....	205
neighbor add-paths advertise-best.....	206
neighbor allowas-in.....	207
neighbor capability orf prefix-list	208
neighbor default-originate.....	209
neighbor ebgp-multihop.....	210
neighbor fall-over.....	211
neighbor fall-over bfd.....	212
neighbor graceful-shutdown.....	213
neighbor local-as.....	215
neighbor maximum-prefix.....	215
neighbor next-hop-self.....	217
neighbor next-hop-unchanged.....	218
neighbor orf prefix-list in.....	219
neighbor passive.....	220
neighbor password.....	220
neighbor port.....	221
neighbor remote-as.....	222
neighbor remove-private-AS.....	223
neighbor route-map.....	223
neighbor route-reflector-client.....	225
neighbor send-community.....	226
neighbor shutdown.....	227
neighbor soft-reconfiguration inbound.....	228
neighbor timers.....	228
neighbor ttl-security-hops.....	229
neighbor update-source.....	230
neighbor weight.....	231
network.....	232
redistribute.....	232
router bgp.....	234
show bgp.....	234
show bgp <PREFIX>.....	236
show bgp community.....	237
show bgp flap-statistics.....	240
show bgp neighbors.....	241
show bgp neighbor advertised-routes.....	245
show bgp neighbor paths.....	246
show bgp neighbor received orf-prefix-list.....	247
show bgp neighbor received-routes.....	248
show bgp neighbor routes.....	250
show bgp neighbors.....	251

show bgp paths.....	254
show bgp peer-group summary.....	255
show bgp summary.....	257
show bgp l2vpn evpn vni route-type.....	259
show bgp l2vpn evpn vtep.....	260
show bgp l2vpn evpn vtep route-type.....	260
show bgp l2vpn evpn vtep vni.....	261
show bgp l2vpn evpn vtep vni route-type.....	262
show running-config bgp.....	263
timers bgp.....	263
vrf.....	264

Chapter 8 Route Policies and Route Maps..... 266

Overview.....	266
Route maps.....	266
Match criteria.....	266
Set changes.....	267
IP prefix lists.....	267
AS-path lists for BGP.....	267
Community lists for BGP.....	267
Route flap dampening.....	268
Route redistribution and route maps.....	268
Route policy and route map commands.....	268
General or filtering commands.....	268
ip aspath-list.....	268
ip community-list.....	269
ip prefix-list.....	270
route-map.....	271
continue.....	272
Match commands.....	273
match aspath-list.....	273
match community-list.....	273
match interface.....	274
match ip address prefix-list.....	275
match ip next-hop.....	275
match ip route-source.....	276
match local-preference.....	277
match metric.....	277
match origin.....	278
match route-type.....	278
match source-protocol.....	279
match tag.....	279
Set commands.....	280
set as-path exclude.....	280
set as-path prepend.....	281
set community.....	281
set dampening.....	282
set ip nexthop.....	283
set local-preference.....	283
set metric.....	284
set origin.....	284
set tag.....	285
set weight.....	286
Show commands.....	286

show ip aspath-list.....	286
show ip community-list.....	287
show ip prefix-list.....	287
show route-map.....	288
Chapter 9 Equal Cost Multipath (ECMP).....	290
Overview.....	290
ECMP commands.....	290
show ip ecmp.....	290
Chapter 10 Virtual Router Redundancy Protocol (VRRP).....	292
Overview.....	292
Terminology.....	292
VRRP operation.....	293
Multiple VRRP groups.....	294
VRRP priority.....	295
VRRP preemption.....	295
Virtual Router MAC address.....	296
VRRP and ARP	296
VRRP and MLAG	296
VRRP tracking.....	296
High availability.....	296
VRRP and Neighbor Discovery for IPv6.....	297
Duplicate address detection (DAD)	297
Guidelines and limitations.....	297
VRRP commands.....	298
address.....	298
authentication.....	299
bfd <IPv4-ADDR>.....	300
preempt.....	301
preempt delay minimum.....	302
priority.....	302
router vrrp enable.....	303
router vrrp disable.....	303
no router vrrp.....	304
show track	304
show track brief.....	305
show vrrp.....	305
shutdown.....	312
timers advertise.....	312
track (VRRP group).....	313
track (VRRP virtual router).....	314
track by.....	314
version.....	315
vrrp.....	315
Chapter 11 Inter-Virtual Router Forwarding (IVRF).....	317
Virtual Routing and Forwarding overview.....	317
Static VRF route leak.....	317
Dynamic VRF route leak.....	317
Procedure to leak routes between nondefault VRFs.....	318
IVRF commands.....	318

address-family.....	318
ip ipv6 vrf.....	319
IPv6 route source interface.....	320
ip route interface.....	322
rd.....	325
redistribute.....	325
route-target.....	327
router bgp.....	328
router bgp vrf.....	328
show bgp vpn unicast.....	329
show bgp info vrf	333
show ip route vrf.....	334
show ipv6 route vrf.....	335
vrf.....	335

Chapter 12 Policy Based Routing (PBR)..... 337

Overview.....	337
PBR actions.....	337
PBR policy action and action list.....	337
PBR action list maximum entries.....	338
IP versions in an action list.....	338
Specifying valid next-hop and default-nexthop addresses.....	339
Hardware path PBR versus software path PBR.....	339
Hardware versus software path for default-nexthop action.....	339
Software path and system default route.....	340
PBR and VRFs.....	340
PBR, ECMP, and routing protocols.....	341
PBR, VSX, and VLAN ACLs.....	341
PBR software path, VSX, and VRRP.....	341
PBR and next-hop router reachability.....	341
CLI errors.....	342
PBR commands.....	342
apply policy.....	342
pbr-action-list.....	343
pbr-action-list copy.....	346
pbr-action-list resequence.....	347
pbr-action-list reset.....	348
policy.....	348
show pbr.....	351
show pbr-action-list.....	352
show running-config current-context.....	353

Chapter 13 IP Directed Broadcast..... 355

IP Directed Broadcast Overview.....	355
IP Directed Broadcast configuration example.....	356
IP Directed Broadcast commands.....	357
copy support-file feature.....	357
ip directed-broadcast.....	357
show ip interface.....	358
show ip directed-broadcast.....	359

Chapter 14 IP Neighbor Flood..... 360

IP Neighbor Flood Overview.....	360
IP Neighbor Flood commands.....	360
ip neighbor-flood.....	360
show ip interface.....	361
show ip neighbor-flood.....	361
show running-config.....	362
Chapter 15 Key chain.....	363
Overview.....	363
Key chain commands.....	363
accept-lifetime.....	363
key.....	364
keychain.....	365
key-string.....	365
send-lifetime.....	366
show capacities keychain.....	367
show keychain.....	368
show running-config keychain.....	369
Chapter 16 IP Client Tracker.....	371
Overview.....	371
IP Client Tracker commands.....	371
client track ip.....	371
client track ip { enable disable auto }.....	372
client track ip client-limit.....	372
client track ip update-interval.....	373
client track ip update-method probe.....	374
show capacities.....	374
show client ip { count port vlan }.....	375
Chapter 17 Routing Information Protocol (RIP).....	376
Overview.....	376
RIPv2 (IPv4) commands.....	376
Configuration commands.....	376
router rip.....	376
Interface commands.....	377
ip rip.....	377
ip rip all-ip enable.....	378
ip rip all-ip disable.....	379
ip rip all-ip send disable.....	379
ip rip all-ip receive disable.....	380
Routing commands.....	381
enable.....	381
disable.....	381
distance.....	382
maximum-paths.....	382
redistribute.....	383
timers update.....	384
RIPv2 clear commands.....	385
clear ip rip statistics.....	385
RIPv2 interface commands.....	385
enable.....	385

disable.....	386
send disable.....	387
receive disable.....	387
RIPv2 show commands.....	388
show capacities rip.....	388
show capacities-status rip.....	389
show ip rip.....	389
show ip rip interface.....	390
show ip rip neighbors.....	391
show ip rip routes.....	392
show ip rip statistics.....	393
show ip rip statistics interface.....	394
show running-config.....	395
RIPng (IPv6) commands.....	397
Configuration commands.....	397
router ripng.....	397
Interface commands.....	397
ipv6 ripng.....	398
Routing commands.....	398
enable.....	398
disable.....	399
distance.....	399
maximum-paths.....	400
redistribute.....	401
timers update.....	401
RIPng clear commands.....	402
clear ipv6 ripng statistics.....	402
RIPng interface commands.....	403
enable.....	403
disable.....	404
send disable.....	404
receive disable.....	405
RIPng show commands.....	405
show capacities ripng.....	405
show capacities-status ripng.....	406
show ipv6 ripng.....	406
show ipv6 ripng interface.....	407
show ipv6 ripng neighbors.....	409
show ipv6 ripng routes.....	410
show ipv6 ripng statistics.....	411
show ipv6 ripng statistics interface.....	412
show running-config.....	413
Chapter 18 Support and other resources.....	415
Accessing Aruba Support.....	415
Accessing updates.....	415
Warranty information.....	416
Regulatory information.....	416
Documentation feedback.....	416

This document describes features of the AOS-CX network operating system. It is intended for administrators responsible for installing, configuring, and managing Aruba switches on a network.

Applicable products

This document applies to the following products:

- Aruba 6300 Switch Series (JL658A, JL659A, JL660A, JL661A, JL662A, JL663A, JL664A, JL665A, JL666A, JL667A, JL668A, JL762A)
- Aruba 6400 Switch Series (JL741A, R0X26A, R0X27A, R0X29A, R0X30A)
- Aruba 8320 Switch Series (JL479A, JL579A, JL581A)
- Aruba 8325 Switch Series (JL624A, JL625A, JL626A, JL627A)
- Aruba 8360 Switch Series (JL700A, JL701A, JL702A, JL703A, JL706A, JL707A, JL708A, JL709A, JL710A, JL711A)

Latest version available online

Updates to this document can occur after initial publication. For the latest versions of product documentation, see the links provided in **Support and other resources**.

Command syntax notation conventions

Convention	Usage
<code>example-text</code>	Identifies commands and their options and operands, code examples, filenames, pathnames, and output displayed in a command window. Items that appear like the example text in the previous column are to be entered exactly as shown and are required unless enclosed in brackets ([]).
example-text	In code and screen examples, indicates text entered by a user.
Any of the following:	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code:
<ul style="list-style-type: none"> • <code><example-text></code> • <code><example-text></code> • <i>example-text</i> • example-text 	<ul style="list-style-type: none"> • For output formats where italic text cannot be displayed, variables are enclosed in angle brackets (< >). Substitute the text—including the enclosing angle brackets—with an actual value. • For output formats where italic text can be displayed, variables might or might not be enclosed in angle brackets. Substitute the text including the enclosing angle brackets, if any, with an actual value.

Table Continued

Convention	Usage
	Vertical bar. A logical OR that separates multiple items from which you can choose only one. Any spaces that are on either side of the vertical bar are included for readability and are not a required part of the command syntax.
{ }	Braces. Indicates that at least one of the enclosed items is required.
[]	Brackets. Indicates that the enclosed item or items are optional.
... or	Ellipsis:
...	<ul style="list-style-type: none"> In code and screen examples, a vertical or horizontal ellipsis indicates an omission of information. In syntax using brackets and braces, an ellipsis indicates items that can be repeated. When an item followed by ellipses is enclosed in brackets, zero or more items can be specified.

About the examples

Examples in this document are representative and might not match your particular switch or environment. The slot and port numbers in this document are for illustration only and might be unavailable on your switch.

Understanding the CLI prompts

When illustrating the prompts in the command line interface (CLI), this document uses the generic term `switch`, instead of the host name of the switch. For example:

```
switch>
```

The CLI prompt indicates the current command context. For example:

```
switch>
```

Indicates the operator command context.

```
switch#
```

Indicates the manager command context.

```
switch(CONTEXT-NAME)#
```

Indicates the configuration context for a feature. For example:

```
switch(config-if) #
```

Identifies the `interface` context.

Variable information in CLI prompts

In certain configuration contexts, the prompt may include variable information. For example, when in the VLAN configuration context, a VLAN number appears in the prompt:

```
switch(config-vlan-100) #
```

When referring to this context, this document uses the syntax:

```
switch(config-vlan-<VLAN-ID>) #
```

Where `<VLAN-ID>` is a variable representing the VLAN number.

Identifying switch ports and interfaces

Physical ports on the switch and their corresponding logical software interfaces are identified using the format:

member/slot/port

On the 6300 Switch Series

- *member*: Member number of the switch in a Virtual Switching Framework (VSF) stack. Range: 1 to 10. The primary switch is always member 1. If the switch is not a member of a VSF stack, then member is 1.
- *slot*: Line module number. Always 1.
- *port*: Physical number of a port on a line module.

For example, the logical interface 1/1/4 in software is associated with physical port 4 in slot 1 on member 1.

On the 6400 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Specifies physical location of a module in the switch chassis.
 - Management modules are on the front of the switch in slots 1/1 and 1/2.
 - Line modules are on the front of the switch starting in slot 1/3.
- *port*: Physical number of a port on a line module.

For example, the logical interface 1/3/4 in software is associated with physical port 4 in slot 3 on member 1.

On the 83xx Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Line module number. Always 1.
- *port*: Physical number of a port on a line module

For example, the logical interface 1/1/4 in software is associated with physical port 4 in slot 1 on member 1.



NOTE: If using breakout cables, the port designation changes to x:y, where x is the physical port and y is the lane when split to 4 x 10G or 4 x 25G. For example, the logical interface 1/1/4:2 in software is associated with lane 2 on physical port 4 in slot 1 on member 1.

Identifying modular switch components

- Power supplies are on the front of the switch behind the bezel above the management modules. Power supplies are labeled in software in the format: *member/power supply*:
 - *member*: 1.
 - *power supply*: 1 to 4.
- Fans are on the rear of the switch and are labeled in software as: *member/tray/fan*:

- *member*: 1.
- *tray*: 1 to 4.
- *fan*: 1 to 4.
- Fabric modules are not labeled on the switch but are labeled in software in the format: *member/module*:
 - *member*: 1.
 - *member*: 1 or 2.
- The display module on the rear of the switch is not labeled with a member or slot number.

Overview

Virtual Routing and Forwarding (VRF) is a Layer 3 level isolation used to achieve Virtual Private Network (VPN). VRF provides overlapping IPs to present and also isolate the routing table from other VPNs in the system.

VRF is a technology that allows multiple instances of a routing table to co-exist within the same router. Because the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other. Network functionality is improved because network paths can be segmented without requiring multiple routers.

Adding or deleting a VRF

Prerequisites

You must be in the global configuration context, as indicated by the `config` prompt.

Procedure

To configure a VRF, enter the following command.

```
vrf <vrf-name>
```

Where `<vrf-name>` is the name of the VRF, up to 32 alphanumeric characters.



CAUTION: The `no` form of the command deletes the VRF and will remove all the configurations from the interfaces which are part of the deleted VRF and move those interfaces to the default VRF.

See the following examples:

```
switch(config) # vrf test
```

```
switch(config) # no vrf test
```

Related topic: [vrf](#) (command reference)

IPv4 static route addition or deletion in a VRF

Prerequisites

You must be in the global configuration context, as indicated by the `config` prompt.

Procedure

To add an IPv4 static route to a VRF, enter the following command.

```
ip route <dest-ipv4-addr>/<netmask> {<gateway-ip>|<interface>} vrf [<vrf-name>]
```

Where

<dest-ipv4-addr>/<netmask>

Specifies the route destination IP address and the network mask length for the destination.

<gateway-ip>|<interface>

Specifies the gateway as either an IP address or an interface.

<vrf-name>

Specifies the VRF name. If no <vrf-name> is specified the route is applied to the default VRF.

Use the `no` form of the command to remove an IPv4 static route from the VRF.

See the following example:

```
switch(config)# ip route 20.0.0.0/24 10.0.0.1 vrf test
switch(config)# ip route 20.0.0.0/24 1/1/5 vrf test
```

```
switch(config)# no ip route 20.0.0.0/24 10.0.0.1 vrf test
```

Related topic: [ip route vrf](#) (command reference)

IPv6 static route addition or deletion in a VRF

Prerequisites

You must be in the global configuration context, as indicated by the `config` prompt.

Procedure

To add an IPv6 static route to a VRF, enter the following command.

```
ipv6 route <dest-ipv6-addr>/<prefix> {<gateway-ip>|<interface>} vrf [<vrf-name>]
```

Where

<dest-ipv6-addr>/<prefix>

Specifies the route destination IPv6 address and the network prefix for the destination. For example, `120::/124`.

<gateway-ip>|<interface>

Specifies either the gateway as either an IPv6 address or an interface.

<vrf-name>

Specifies the VRF name. The name can be up to 32 alphanumeric characters. If no <vrf-name> is specified the route is applied to the default VRF.

Use the `no` form of the command to delete an IPv6 static route from a VRF.

See the following examples:

```
switch(config)# ipv6 route 120::/124 121::2 vrf test
switch(config)# ipv6 route 120::/124 1/1/9 vrf test
```

```
switch(config)# no ipv6 route 120::/124 121::2 vrf test
```

Related topics: [ipv6 route vrf](#) (command reference)

Attaching or detaching a port from a VRF

Prerequisites

You must be in the interface configuration context (`config-if`), or the VLAN interface configuration context (`config-vlan-if`), or the LAG interface configuration context (`config-lag-if`).

Procedure

To attach a port to a VRF, enter the following command in the required context.

```
vrf attach <vrf-name>
```

Where `<vrf-name>` is the name of the VRF, up to 32 alphanumeric characters.



CAUTION: The `no` form of the command detaches the port from the named VRF and will remove all configurations from the port and attach the port to the default VRF.

See the following examples:

```
switch(config)# interface 1/1/1
switch(config-if)# vrf attach test
```

```
switch(config)# vlan 3
switch(config-vlan)# exit
switch(config)# interface vlan 3
switch(config-if-vlan)# vrf attach test
```

Related topic: [vrf attach](#) (command reference)

Viewing VRF information

Prerequisites

These commands are in the switch context, executed at the `switch#` prompt.

Procedure

1. To view various aspects of VRF information, use the following commands.

- To view VRF configuration and status information:

```
show vrf <vrf-name>
```
- To view all configured commands including VRF configuration:

```
show running-config
```

For command details and examples, see the following:

- show vrf
- An example of the VRF information provided by the show running-config command

An example of the VRF information provided by the show running-config command

When a VRF is configured, the output of the `show running-config` command includes information about the VRF configuration.

Example

```
switch# show running-config
Current configuration:
!
!Version ArubaOS-CX 10.0X.XXXX
!
lldp enable
timezone set utc
vrf new
vrf test
!
!
!
aaa authentication login default local
aaa authorization commands default none
!
!
!
vlan 1
    no shutdown
interface 1/1/1
    no shutdown
    vrf attach test
    ip address 20.0.0.2/24
    ipv6 address 120::1/124
interface 1/1/2
    no shutdown
    vrf attach new
    ip address 30.0.0.1/24
    ipv6 address 121::1/124
```

VRF commands

ip route vrf

Syntax

```
ip route <DEST-IPV4-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
no ip route <DEST-IPV4-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

Description

Adds the destination IPv4 static route on the specified VRF. If no `<VRF-NAME>` is specified the route is applied to the default VRF.

The `no` form of this command removes the IPv4 static route from the VRF.

Command context

config

Parameters

`<DEST-IPV6-ADDR>`

Specifies the route destination in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

`<MASK>`

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

`<NEXT-HOP-IP-ADDR>`

Specifies the next hop in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

`<INTERFACE>`

Specifies the next hop as an outgoing interface.

blackhole

Specifies that packets matching the destination prefix are silently discarded and no ICMP error notification is sent to the sender.

reject

Specifies that packets matching the destination prefix are discarded and an ICMP error notification is sent to the sender.

`vrf <VRF-NAME>`

Specifies a VRF name.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ip route 20.0.0.0/8 10.20.30.44 vrf myvrf
switch(config)# ip route 20.1.2.0/24 1/1/30 vrf myvrf
switch(config)# ip route 1.2.3.4/32 blackhole vrf myvrf
switch(config)# ip route 1.2.3.4/32 reject vrf myvrf
```

```
switch(config)# no ip route 20.0.0.0/8 10.20.30.44 vrf myvrf
```

ipv6 route vrf

Syntax

```
ipv6 route <DEST-IPV6-ADDR>/<PREFIX> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

```
no ipv6 route <DEST-IPV6-ADDR>/<PREFIX> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

Description

Adds an IPv6 static route in the specified VRF. If no `<VRF-NAME>` is specified it is added to the default VRF. The `no` form of this command removes an IPv6 static route from the VRF.

Command context

config

Parameters

`<DEST-IPV6-ADDR>`

Specifies an IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

`<MASK>`

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

`<NEXT-HOP-IP-ADDR>`

Specifies the next hop in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

`<INTERFACE>`

Specifies the next hop as an outgoing interface.

`blackhole`

Specifies that packets matching the destination prefix are silently discarded and no ICMP error notification is sent to the sender.

`reject`

Specifies that packets matching the destination prefix are discarded and an ICMP error notification is sent to the sender.

`vrf <VRF-NAME>`

Specifies the name of a VRF. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ipv6 route 120::/124 121::2 vrf test
switch(config)# ipv6 route 121::/124 1/1/9 vrf test
switch(config)# ipv6 route 122::/124 blackhole vrf test
switch(config)# ipv6 route 123::/124 reject vrf test
```

```
switch(config)# no ipv6 route 120::/124 121::2 vrf test
```

show ip route

Syntax

```
show ip route [<A.B.C.D> | <A.B.C.D/M> | all-vrfs | bgp | connected | local | ospf | static | summary |
               vrf<vrf-name>] [vsx-peer]
```

Description

Displays IPv4 route tables.

Command context

Operator (>) or Manager (#)

Parameters

<A.B.C.D>

Display longest prefix match.

<A.B.C.D/M>

Display exact route match.

all-vrfs

Display information for all VRFs.

bgp

Display bgp routes only.

connected

Display connected routes only.

local

Display local routes only.

ospf

Display ospf routes only.

static

Display static routes only.

summary

Display the aggregate count of routes per routing protocol.

vrf <vrf-name>

Specify a VRF by VRF name (if no *<VRF-NAME>* is specified, the default VRF is implied).

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing IPv4 route tables:

```
switch# show ip route
```

Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]


```
10.0.0.0/24, 1 (null) next-hops
    via 10.0.0.1, [0/0], connected
20.0.0.0/24, 1 unicast next-hops
    via 10.0.0.2, [1/0], static
20.0.1.0/24, 1 unicast next-hops
    via 10.0.0.2, [3/0], static
20.0.2.0/24, 1 unicast next-hops
    via 1/1/1, [1/0], static
20.0.3.0/24, 1 unicast next-hops
    via 1/1/1, [4/0], static
20.0.4.0/24, blackhole, 1, [1/0], static
20.0.5.0/24, reject, 1, [1/0], static
```

Showing IPv4 route tables for the test VRF:

```
switch# show ip route vrf test

Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

30.0.0.0/24, 1 (null) next-hops
    via 30.0.0.2, [0/0], connected
90.0.0.0/24, 1 unicast next-hops
    via 30.0.0.1, [1/0], static
90.0.1.0/24, 1 unicast next-hops
    via 1/1/2, [1/0], static
90.0.3.0/24, blackhole, 1, [1/0], static
```

show ipv6 route

Syntax

```
show ipv6 route [<X.X.X.X> | <X.X.X.X/M> | all-vrfs | bgp | connected | local | ospf | static | summary |
vrf <vrf-name>] [vsx-peer]
```

Description

Displays IPv6 route tables.

Command context

Operator (>) or Manager (#)

Parameters

<X.X.X.X>

Display exact route match.

<X.X.X.X/M>

Display exact route match.

all-vrfs

Display information for all VRFs.

bgp

Display bgp routes only.

connected

Display connected routes only.

local

Display local routes only.

ospf

Display ospf routes only.

static

Display static routes only.

summary

Display the aggregate count of routes per routing protocol.

vrf <vrf-name>

Specify a VRF by VRF name (if no *<VRF-NAME>* is specified, the default VRF is implied).

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Showing IPv6 route tables:

```
switch# show ipv6 route
```

Displaying ipv6 routes selected for forwarding

'[x/y]' denotes [distance/metric]

```
110::/124, 1 unicast next-hops
    via 120::1, [1/0], static
111::/124, 1 unicast next-hops
    via 1/1/1, [1/0], static
112::/124, 1 unicast next-hops
    via 1/1/1, [3/0], static
113::/124, 1 unicast next-hops
    via 120::1, [5/0], static
114::/124, blackhole, 1, [1/0], static
115::/124, reject, 1, [1/0], static
120::/124, 1 (null) next-hops
    via 120::2, [0/0], connected
```

show vrf**Syntax**

```
show vrf <VRF-NAME>
```

```
show vrf
```

Description

Displays the status and attached interfaces for the specified VRF instance.

The `show vrf` command shows this information for all the VRFs.

Command context

Operator (>) or Manager (#)

Parameters

<VRF-NAME>

Specifies the VRF name. Length: Up to 32 alphanumeric characters.

Authority

Operators or Administrators or local user group members with execution rights for this command.

Operators can execute this command from the operator context (>) only.

Examples

Showing VRF information for the test VRF:

```
switch# show vrf test
VRF Configuration:
-----
VRF Name      : test
  Interfaces              Status
  -----
  1/1/29                up
  1/1/30                up
```

Showing VRF information for all VRFs:

```
switch# show vrf
VRF Configuration:
-----
VRF Name      : default
  Interfaces              Status
  -----

VRF Name      : red
  Interfaces              Status
  -----
  1/1/32                up

VRF Name      : test
  Interfaces              Status
  -----
  1/1/29                up
  1/1/30                up
```

vrf

Syntax

`vrf <VRF-NAME>`

`no vrf <VRF-NAME>`

Description

This command configures a named VRF instance `<VRF-NAME>`. After creating the VRF instance, the command prompt enters into the VRF configure context.

The `no` form of the command deletes the VRF instance from the router.

Command context

```
config
```

Parameters

`<VRF-NAME>`

Required: Specifies the VRF name. Length: Up to 32 alphanumeric characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring VRF for a named customer:

```
switch(config)# vrf cust_A
```

Removing the VRF configuration with the name `test`:

```
switch(config)# no vrf test
```

vrf attach

Syntax

```
vrf attach <VRF-NAME>
```

```
no vrf attach <VRF-NAME>
```

Description

Attaches the interface to the VRF with the name `<VRF-NAME>`. The command can be entered in several different command contexts.



CAUTION: The `no` form of the command detaches the interface from the named VRF and will remove all configurations from the interface and attach the interface to the default VRF. A warning message is displayed that prompts you whether to proceed: All Layer 3 configurations associated with the VRF will be deleted. Continue (y/n)?

Command context

```
config-if
```

```
config-if-vlan
```

```
config-lag-if
```

```
config-gre-if
```

Parameters

`<VRF-NAME>`

Specifies the VRF name. Required. Length: Up to 32 alphanumeric characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/29  
switch(config-if)# vrf attach test
```

```
switch(config)# vlan 3  
switch(config-vlan)# exit  
switch(config)# interface vlan 3  
switch(config-if-vlan)# vrf attach test
```

```
switch(config)# vrf test  
switch(config)# interface lag 3  
switch(config-lag-if)# no shutdown  
switch(config-lag-if)# vrf attach test
```

```
switch(config)# interface 1/1/29  
switch(config-if)# no vrf attach test
```

Overview

A loopback interface is a virtual interface supporting IPv4/IPv6 address configuration. The loopback interface can be considered stable because once created, it remains up. The loopback interface can then be configured with an address to use as a reference or identifier independent of the physical interfaces.

- **Device identification:** As long as the router is operational, the state of the loopback interface is always up. Even if only one link to the router is active, the loopback interface can be reached. This functionality makes it possible to identify an active device in the network using the IP address configured on the loopback interface.
- **Routing:** Since the loopback interface is always active, a routing session (such as a BGP session) can continue on an alternate path even if the outbound interface fails. In OSPF, a loopback interface address is advertised as an interface route into the network. This functionality increases reliability by allowing traffic to take alternate paths if there is a link failure. In OSPF and BGP, the router ID can be set to the loopback address to avoid reassignment of the router ID when physical interfaces are added or removed.
- **Device management:** Loopback interface is always reachable and can be used for sending and receiving management information such as logs and SNMP traps without interruption.

Loopback commands

interface loopback

Syntax

```
interface loopback <INSTANCE>
```

```
no interface loopback <INSTANCE>
```

Description

Creates a loopback interface and enters loopback configuration mode.

The `no` form of this command deletes a loopback interface.

Command context

```
config
```

Parameters

<INSTANCE>

Selects the loopback interface ID. Range: 0 to 255

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface loopback 1  
switch(config-loopback-if)#
```

ip address

Syntax

```
ip address <IPV4-ADDR/MASK> [secondary]
```

```
no ip address <IPV4-ADDR/MASK> [secondary]
```

Description

Sets the IPv4 address for a loopback interface.

The `no` form of this command reverses the set of the IPv4 address for a loopback interface.

Command context

config

Parameters

<IPV4-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

secondary

Indicates that the IPv4 address is a secondary address.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface loopback 1  
switch(config-loopback-if)# ip address 16.93.50.2/24  
switch(config-loopback-if)# ip address 20.1.1.1/24 secondary
```

ipv6 address

Syntax

```
ipv6 address <IPV6-ADDR/MASK>
```

Description

Sets the IPv6 address for a loopback interface.

Command context

config

Parameters

<IPV6-ADDR>

Specifies an IP address in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface loopback 1
switch(config-loopback-if)# ipv6 address fd00:5708::f02d:4df6/64
```

vrf attach <VRF-NAME>

Syntax

```
vrf attach <VRF-NAME>
```

```
no vrf attach <VRF-NAME>
```

Description

Attaches a non-default VRF to a loopback.

The no form of this command deletes a non-default VRF from a loopback and reattaches the default VRF.

Command context

```
config
```

Parameters

<VRF-NAME>

Specifies the name of the non-default VRF to be attached/deleted to/from a loopback.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface loopback 1
switch(config-loopback-if)#vrf attach test
```

show interface loopback

Syntax

```
show interface loopback [brief | instance <ID>] [vsx-peer]
```


Description

This command displays the configuration and status of loopback interfaces.

Command context

Operator (>) or Manager (#)

Parameters

brief

Displays brief information about all configured loopback interfaces.

instance <ID>

Displays the configuration and status of a loopback interface ID. Range: 1-255

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show interface loopback
```

```
Interface loopback1 is up
IPv4 address 192.168.1.1/24
```

```
Interface loopback2 is up
IPv4 address 182.168.1.1/24
```

```
switch# show interface loopback brief
```

Loopback Interface	IP Address	Status
loopback1	10.1.1.1/24	up
loopback1	1111:2222:3333:4444::6666/128	up

```
switch# show interface loopback 1
```

```
Interface loopback1 is up
IPv4 address 192.168.1.1/24
```

Overview

Static routes are manually configured. If the network topology is simple enough, you can use static routes for the network's routing requirements. The proper configuration and usage of static routes can improve network performance and ensure bandwidth for important network applications.

The disadvantage of using static routes is that they cannot adapt to network topology changes. If a fault or topological change occurs in the network, the relevant routes will be unreachable and a network administrator must modify the static routes manually.

Default route

Without a default route, a packet that does not match any routing entries is discarded and an ICMP destination-unreachable packet is sent to the source. A default route is used to forward packets that do not match any routing entry.

The network administrator can configure a default route with the destination as 0.0.0.0 and the mask as 0. The router forwards any packet whose destination address fails to match any entry in the routing table to the next hop of the default static route.

Recursive static routes

Recursive static routes are not supported.

A recursive static route is a route for which the next hop is learned from another routing look up (for example, dynamic protocol or from another another static route). For example, the following commands create an unsupported recursive static route:

- `ip route 99.0.0.0/24 30.0.0.2` - Main static route with gateway 30.0.0.2.
- `ip route 30.0.0.0/24 20.0.0.2` - Static route to reach the nexthop of the previously configured route.

Configuration concepts

Before configuring a static route, you must understand the following concepts:

- **Destination address and mask:** In the `ip route` command, an IPv4 address is in dotted-decimal format. A mask can be in the form of mask length - the number of consecutive 1s in the mask.
- **Output interface and next hop address:** When configuring a static route, specify the output interface or next hop address. The next hop address cannot be a local interface IP address or the route configuration will not take effect.
- **Other attributes:** You can configure different priorities and administrative distance for different static routes to make route management policies more flexible. For example, specifying the same priority for different routes to the same destination enables load sharing, but specifying different priorities for these routes enables route backup.

Configuration example procedure

Configuration prerequisites

Before configuring a static route, complete the following tasks:

- Configure the physical parameters for related interfaces.
- Configure the link-layer attributes for related interfaces.
- Configure the IP addresses for related interfaces.



NOTE: The number of IPv4 and IPv6 static routes that can be configured is limited to 16K combined.

For the 6300 and 6400 switches, the number of route prefixes can be stored in the routing table, depending on the prefix mask length (data set) being used (for example, /8, /16, /24, etc.). These switches do not support IPv6 addresses with prefixes greater than 64, and for prefixes between 65 and 127 will be software-forwarded.

Basic static route configuration example

The IP addresses and masks of the switches and hosts are displayed here. Static routes are required for interconnection between any two hosts.

Procedure

1. Configure IP addresses for interfaces (details not shown).

2. Configure static routes.

a. Configure a default route on Switch A.

```
6300# config
6300(config)#
<SwitchA> config
[SwitchA] ip route 0.0.0.0/0 1.1.4.2
```

b. Configure two static routes on Switch B.

```
<SwitchB> config
[SwitchB] ip route 1.1.2.0/24 1.1.4.1
[SwitchB] ip route 1.1.3.0/24 1.1.5.6
```

c. Configure a default route on Switch C.

```
6300# config
6300(config)#
[SwitchC] ip route 0.0.0.0/0 1.1.5.5
```

3. Configure the hosts. The default gateways for hosts A, B, and C are 1.1.2.3, 1.1.6.1, and 1.1.3.1. The configuration procedure is not shown.

4. Display the configuration.

a. Display the IP routing table of Switch A.

b. Display the IP routing table of Switch B.

- c. Use the `ping` command on Host B to check the reachability of Host A, assuming Windows XP runs on the two hosts.

```
C:\Documents and Settings\Administrator>ping 1.1.2.2
Pinging 1.1.2.2 with 32 bytes of data:
Reply from 1.1.2.2: bytes=32 time=1ms TTL=255
Reply from 1.1.2.2: bytes=32 time=1ms TTL=255
Reply from 1.1.2.2: bytes=32 time=1ms TTL=255
Reply from 1.1.2.2: bytes=32 time=1ms TTL=255
15
Ping statistics for 1.1.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

- d. Use the `tracert` command on Host B to check the reachability of Host A.

```
8320# tracert
tracert Trace the IPv4 route to a device on the network
tracert6 Trace the IPv6 route to a device on the network
8320# tracert
A.B.C.D Enter IPv4 address of the device to tracert
WORD Enter host name of the device to tracert
8320# tracert
```

Static routing commands

ip route

Syntax

```
ip route <DEST-IPV4-ADDR>/<NETMASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole]
no ip route <DEST-IPV4-ADDR>/<NETMASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole]
```

Description

Adds an IPv4 static route on the default VRF.

The `no` form of this command deletes a IPv4 static route.

Command context

`config`

Parameters

<DEST-IPV4-ADDR>/<NETMASK>

Specifies the IPv4 route destination.

<NEXT-HOP-IP-ADDR>

Specifies the next hop address for reaching the destination in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<INTERFACE>

Specifies the next hop as an outgoing interface.

blackhole

Specifies that packets matching the destination route are silently discarded and no ICMP error notification is sent to the sender.

reject

Specifies that packets matching the destination route are discarded and an ICMP error notification is sent to the sender.

Authority

Administrators or local user group members with execution rights for this command.

Examples

On the 6400 Switch Series, interface identification differs.

```
switch(config)# ip route 10.0.0.0/24 blackhole
switch(config)# ip route 10.0.1.0/24 reject
switch(config)# ip route 10.0.2.0/24 20.0.0.2
switch(config)# ip route 10.0.3.0/24 1/1/1
```

ip route bfd

Syntax

```
ip route <DEST-IPV4-ADDR>/<NETMASK> [<NEXT-HOP-IP-ADDR> | <INTERFACE>] [bfd]
```

Description

Enables or disables BFD on the specified static route. To disable BFD, issue the command without the `bfd` option.

Command context

config

Parameters

<DEST-IPV4-ADDR>

Specifies a route destination in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

<NETMASK>

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

<NEXT-HOP-IP-ADDR>

Specifies the next hop address for reaching the destination in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

<INTERFACE>

Specifies the next hop as an outgoing interface.

bfd

Enables BFD on the static route. Omit this parameter to disable BFD.

Authority

Administrators or local user group members with execution rights for this command.

Examples

On the 6400 Switch Series, interface identification differs.

Enabling BFD on a static route:

```
switch(config)# interface 1/1/1
switch(config-if)# ip address 20.1.1.2/24
switch(config-if)# no shutdown
switch(config-if)# routing
switch(config-if)# exit
switch(config)# ip route 192.0.0.0/8 20.1.1.1 bfd
```

Disabling BFD on a static route:

```
switch(config)# ip route 192.0.0.0/8 20.1.1.1
```

Enabling BFD on a static route:

```
switch(config)# interface 1/1/1
switch(config-if)# ip address 20.1.1.2/24
switch(config-if)# no shutdown
switch(config-if)# routing
switch(config-if)# exit
switch(config)# ip route 192.0.0.0/8 1/1/1 bfd
```

Disabling BFD on a static route:

```
switch(config)# ip route 192.0.0.0/8 1/1/1
```

ip route distance

Syntax

```
ip route <DEST-IPV4-ADDR>/<NETMASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>] distance <VALUE>
no ip route <DEST-IPV4-ADDR>/<NETMASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>] distance <VALUE>
```

Description

Configures the administrative distance for the IPv4 static route.

The **no** form of this command deletes the static route.

Command context

config

Parameters

<DEST-IPV4-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 32.

<NEXT-HOP-IP-ADDR>

Specifies the next hop IPv4 address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<INTERFACE>

Specifies the next hop as an outgoing interface.

distance <VALUE>

Specifies the administrative distance to associate with this static route. Default: 1. Range: 1-255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ip route 10.0.2.0/24 20.0.0.2 distance 4
switch(config)# ip route 10.0.3.0/24 1/1/1 distance 6
```

```
switch(config)# no ip route 10.0.3.0/24 1/1/1 distance 6
```

ip route tag

Syntax

```
ip route <A.B.C.D/M> [A.B.C.D | INTERFACE] tag <1-4294967295>
```

```
no ip route <A.B.C.D/M> [A.B.C.D | INTERFACE] tag <1-4294967295>
```

Description

Configures tag for IPv4 static route.

The `no` form of this command deletes tag for IPv4 static route.

Command context

config

Parameters

tag

Specifies and assigns tag for the route.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ip route 10.1.1.1/32 20.1.1.2 tag 10
switch(config)# ip route 10.1.1.5/32 1/1/1 tag 20
```

```
switch(config)# no ip route 10.1.1.1/32 20.1.1.2 tag 10
switch(config)# no route 10.1.1.5/32 1/1/1 tag 20
```

ip route vrf

Syntax

```
ip route <DEST-IPV4-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

```
no ip route <DEST-IPV4-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

Description

Adds the destination IPv4 static route on the specified VRF. If no `<VRF-NAME>` is specified the route is applied to the default VRF.

The `no` form of this command removes the IPv4 static route from the VRF.

Command context

config

Parameters

`<DEST-IPV6-ADDR>`

Specifies the route destination in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

`<MASK>`

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

`<NEXT-HOP-IP-ADDR>`

Specifies the next hop in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

`<INTERFACE>`

Specifies the next hop as an outgoing interface.

blackhole

Specifies that packets matching the destination prefix are silently discarded and no ICMP error notification is sent to the sender.

reject

Specifies that packets matching the destination prefix are discarded and an ICMP error notification is sent to the sender.

vrf `<VRF-NAME>`

Specifies a VRF name.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ip route 20.0.0.0/8 10.20.30.44 vrf myvrf
switch(config)# ip route 20.1.2.0/24 1/1/30 vrf myvrf
switch(config)# ip route 1.2.3.4/32 blackhole vrf myvrf
switch(config)# ip route 1.2.3.4/32 reject vrf myvrf
```

```
switch(config)# no ip route 20.0.0.0/8 10.20.30.44 vrf myvrf
```

ipv6 route

Syntax

```
ipv6 route <DEST-IPV6-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>]|reject|blackhole]
```

```
no ipv6 route <DEST-IPV6-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>]|reject|blackhole]
```


Description

Adds an IPv6 static route.

The `no` form of this command deletes an IPv6 static route on the default VRF.

Command context

`config`

Parameters

<DEST-IPV6-ADDR>

Specifies the route destination in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

<NEXT-HOP-IP-ADDR>

Specifies the next hop in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<INTERFACE>

Specifies the next hop as an outgoing interface.

blackhole

Specifies that packets matching the destination route are silently discarded and no ICMP error notification is sent to the sender.

reject

Specifies that packets matching the destination route are discarded and an ICMP error notification is sent to the sender.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Access platforms for IPv6 routes above /64 are not supported.
- IPv6 addresses with prefixes 65-127 will not be configured in the ASIC route table.

Examples

On the 6400 Switch Series, interface identification differs.

```
switch(config)# ipv6 route 120::/124 blackhole
switch(config)# ipv6 route 121::/124 reject
switch(config)# ipv6 route 122::/124 1/1/1
switch(config)# ipv6 route 123::/124 120::1
```

```
switch(config)# no ipv6 route 122::/124 1/1/1
```

ipv6 route distance

Syntax

```
ipv6 route <DEST-IPV6-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>] distance <VALUE>  
no ipv6 route <DEST-IPV6-ADDR>/<MASK> [<NEXT-HOP-IP-ADDR>|<INTERFACE>] distance <VALUE>
```

Description

Configures the administrative distance for the IPv6 static route

The **no** form of this command deletes the static route.

Command context

config

Parameters

<DEST-IPV6-ADDR>

Specifies the route destination address in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

<NEXT-HOP-IP-ADDR>

Specifies the next hop in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<INTERFACE>

Specifies the next hop as an outgoing interface.

distance <VALUE>

Specifies the administrative distance to associate with this static route. Range: 1 to 255. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

On the 6400 Switch Series, interface identification differs.

```
switch(config)# ipv6 route 122::/124 1/1/1 distance 5  
switch(config)# ipv6 route 123::/124 120::1 distance 6  
  
switch(config)# no ipv6 route 123::/124 120::1 distance 6
```

ipv6 route tag

Syntax

```
ipv6 route <X:X::X:X/M> [X:X::X:X | INTERFACE] tag <1-4294967295>  
no ipv6 route <X:X::X:X/M> [X:X::X:X | INTERFACE] tag <1-4294967295>
```

Description

Configures tag for IPv6 static route.

Command context

config

Parameters

tag

Specifies and assigns tag for the route.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ipv6 route 3001::1/128 1/1/1 tag 10
switch(config)# ipv6 route 3002::1/128 1000::2 tag 20
```

```
switch(config)# no ipv6 route 3001::1/128 1/1/1 tag 10
switch(config)# no ipv6 route 3002::1/128 1000::2 tag 20
```

ipv6 route vrf

Syntax

```
ipv6 route <DEST-IPV6-ADDR>/<PREFIX> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
no ipv6 route <DEST-IPV6-ADDR>/<PREFIX> [<NEXT-HOP-IP-ADDR>|<INTERFACE>|reject|blackhole] vrf <VRF-NAME>
```

Description

Adds an IPv6 static route in the specified VRF. If no <VRF-NAME> is specified it is added to the default VRF.

The **no** form of this command removes an IPv6 static route from the VRF.

Command context

config

Parameters

<DEST-IPV6-ADDR>

Specifies an IP address in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

<NEXT-HOP-IP-ADDR>

Specifies the next hop in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<INTERFACE>

Specifies the next hop as an outgoing interface.

blackhole

Specifies that packets matching the destination prefix are silently discarded and no ICMP error notification is sent to the sender.

reject

Specifies that packets matching the destination prefix are discarded and an ICMP error notification is sent to the sender.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# ipv6 route 120::/124 121::2 vrf test
switch(config)# ipv6 route 121::/124 1/1/9 vrf test
switch(config)# ipv6 route 122::/124 blackhole vrf test
switch(config)# ipv6 route 123::/124 reject vrf test
```

```
switch(config)# no ipv6 route 120::/124 121::2 vrf test
```

show ip route

Syntax

```
show ip route [<A.B.C.D> | <A.B.C.D/M> | all-vrfs | bgp | connected | local | ospf | static | summary |
               vrf<vrf-name>] [vsx-peer]
```

Description

Displays IPv4 route tables.

Command context

Operator (>) or Manager (#)

Parameters

<A.B.C.D>

Display longest prefix match.

<A.B.C.D/M>

Display exact route match.

all-vrfs

Display information for all VRFs.

bgp

Display bgp routes only.

connected

Display connected routes only.

local

Display local routes only.

ospf

Display ospf routes only.

static

Display static routes only.

summary

Display the aggregate count of routes per routing protocol.

vrf <vrf-name>

Specify a VRF by VRF name (if no *<VRF-NAME>* is specified, the default VRF is implied).

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing IPv4 route tables:

```
switch# show ip route

Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

10.0.0.0/24, 1 (null) next-hops
    via 10.0.0.1, [0/0], connected
20.0.0.0/24, 1 unicast next-hops
    via 10.0.0.2, [1/0], static
20.0.1.0/24, 1 unicast next-hops
    via 10.0.0.2, [3/0], static
20.0.2.0/24, 1 unicast next-hops
    via 1/1/1, [1/0], static
20.0.3.0/24, 1 unicast next-hops
    via 1/1/1, [4/0], static
20.0.4.0/24, blackhole, 1, [1/0], static
20.0.5.0/24, reject, 1, [1/0], static
```

Showing IPv4 route tables for the test VRF:

```
switch# show ip route vrf test

Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

30.0.0.0/24, 1 (null) next-hops
    via 30.0.0.2, [0/0], connected
90.0.0.0/24, 1 unicast next-hops
    via 30.0.0.1, [1/0], static
90.0.1.0/24, 1 unicast next-hops
```

```
via 1/1/2, [1/0], static  
90.0.3.0/24, blackhole, 1, [1/0], static
```

show ipv6 route

Syntax

```
show ipv6 route [<X.X.X.X> | <X.X.X.X/M> | all-vrfs | bgp | connected | local | ospf | static | summary |  
vrf <vrf-name>] [vsx-peer]
```

Description

Displays IPv6 route tables.

Command context

Operator (>) or Manager (#)

Parameters

<X.X.X.X>

Display exact route match.

<X.X.X.X/M>

Display exact route match.

all-vrfs

Display information for all VRFs.

bgp

Display bgp routes only.

connected

Display connected routes only.

local

Display local routes only.

ospf

Display ospf routes only.

static

Display static routes only.

summary

Display the aggregate count of routes per routing protocol.

vrf <vrf-name>

Specify a VRF by VRF name (if no *<VRF-NAME>* is specified, the default VRF is implied).

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Showing IPv6 route tables:

```
switch# show ipv6 route
```

Displaying ipv6 routes selected for forwarding

'[x/y]' denotes [distance/metric]

```
110::/124, 1 unicast next-hops  
    via 120::1, [1/0], static  
111::/124, 1 unicast next-hops  
    via 1/1/1, [1/0], static  
112::/124, 1 unicast next-hops  
    via 1/1/1, [3/0], static  
113::/124, 1 unicast next-hops  
    via 120::1, [5/0], static  
114::/124, blackhole, 1, [1/0], static  
115::/124, reject, 1, [1/0], static  
120::/124, 1 (null) next-hops  
    via 120::2, [0/0], connected
```

Overview

Open Shortest Path First version 2 (OSPFv2) is a routing protocol described in RFC2328. It is a Link State-based IGP (Interior Gateway Protocol) routing protocol applied to routers grouped into OSPF areas identified by the routing configuration on each routing switch. It is widely used in medium to large-sized enterprise networks.

The characteristics of OSPFv2 are:

- Provides a loop-free topology using SPF algorithm.
- Allows hierarchical routing using area 0 (backbone area) as the top of the hierarchy.
- Supports load balancing with equal cost routes for the same destination.
- OSPFv2 is a classless protocol and allows for a hierarchical design with VLSM (Variable Length Subnet Masking) and route summarization.
- Provides authentication of routing messages.
- Scales easily using the concept of OSPF areas.
- Provides fast convergence with triggered, incremental updates via LSAs.

Some OSPFv2 configuration is done in the global configuration context, others in the router ospf context, or in the interface configuration context, or in the vlink context. OSPFv2 can be configured on L3 ports, VLAN interfaces, LAG interfaces, and loopback interfaces. All such configurations work in the mentioned interfaces context. OSPFv2 mandates the associated interface to be a routed interface.

Supported features

- OSPFv2 neighbor adjacency, Hello protocol, multiple areas, Inter-area routing
- AS external routes, stub areas, totally stubby areas, NSSA, ABR, ASBR
- Designated Router/Backup Designated Router
- Point-to-point interfaces/broadcast interfaces
- Virtual Links
- Bidirectional Forwarding Detection (BFD) - refer to the *High Availability Guide* for additional information
- Equal-cost multipath
- Null authentication, Simple password authentication, MD5 authentication
- Area range aggregation - Type-3/Type-7
- External route aggregation - Type-5 LSAs
- Graceful Restart - un-planned, restart interval, Graceful Restart Helper
- Stub router advertisement
- SPF throttling

- LSA Throttling
- SPF throttling
- Configuration of interface parameters such as priority, cost, hello-interval, dead-interval, retransmit-interval, transit-delay, etc.
- Configuration of virtual link parameters such as hello-interval, dead-interval, retransmit-interval, transit-delay, etc.
- Route redistribution
- Passive interfaces
- Multi VRF support with each VRF having up to eight OSPF process instances
- Congestion control (prioritizing hello packets, inactivity timer reset, and adjacency throttling)

How OSPFv2 protocol works

OSPF protocol

The protocol uses Link State Advertisements (LSAs) transmitted by each router to update neighboring routers regarding its interfaces and the routes available through those interfaces. Each routing switch in an area also maintains a link-state database (LSDB) that describes the area topology. (All routers in a given OSPF area have identical LSDBs.) The routing switches used to connect areas to each other flood summary link LSAs and external link LSAs to neighboring OSPF areas to update them regarding available routes. Through this means, each OSPF router determines the shortest path between itself and a desired destination router in the same OSPF domain (Autonomous System (AS)).

OSPFv2 concepts

OSPFv2 Link-state advertisement (LSA) types

OSPFv2 sends routing information in Link-State Advertisements (LSAs). The switches support the following types of LSAs.

- **Router LSA—Type-1 LSA.** Describes the states of the router interfaces to an area, and is flooded throughout a single area only.
- **Network LSA—Type-2 LSA.** Describes the list of routers connected to the network, and is flooded throughout a single area only.
- **Summary LSA—Type-3 LSA.** Describes the route to networks in another OSPF area of the same Autonomous System (AS). Propagated through backbone area to other areas.
- **ASBR summary LSA—Type-4 LSA.** Describes the route to an ASBR in an OSPF normal or backbone area of the same AS. Propagated through backbone area to other areas.
- **AS external LSA—Type-5 LSA.** Describes the route to a destination in another AS (external route.) Originated by ASBR in normal or backbone areas of an AS and propagates through backbone area to other normal areas.
- **NSSA LSA—Type-7 LSA.** Describes the route to a destination in another AS (external route.) Originated by ASBR in NSSA. ABR converts type-7 LSAs to type-5 LSAs for injection into the backbone area.

OSPFv2 router types

Internal routers

Internal OSPFv2 routers belong to only one area. Internal routers flood type-1 LSAs to all routers in the same area and maintain identical LSDBs.

Area border routers (ABRs)

Area border routers have membership in multiple areas. ABRs are used to connect the various areas in an AS to the backbone area for that AS. Multiple ABRs can be used to connect a given area to the backbone, and a given ABR can belong to multiple areas other than the backbone.

An ABR maintains a separate LSDB for each area to which it belongs. (All routers within the same area have identical LSDBs.) The ABR is responsible for flooding summary LSAs between its border areas. You can reduce summary LSA flooding by configuring area ranges. An area range enables you to assign an aggregate address to a range of IP addresses. This aggregate address is advertised instead of all the individual addresses it represents.

Autonomous system boundary router (ASBR)

Autonomous system boundary routers run multiple interior gateway protocols and serve as a gateway to other autonomous systems operating with interior gateway protocols. The ASBR imports and translates different protocol routes into OSPF through redistribution. ASBRs can be used in backbone areas, normal areas, and NSSAs, but not in stub areas.

Designated routers (DRs)

In an OSPF network having two or more routers, one router is elected to serve as the DR and another router to act as the Backup Designated Router (BDR). All other routers in the area forward their routing information to the DR and BDR, and the DR forwards this information to all routers in the network. This action minimizes the amount of repetitive information that is forwarded on the network by eliminating the need for each individual router in the area to forward its routing information to all other routers in the network. If the area includes multiple networks, each network elects its own DR and BDR.

In an OSPF network with no DR and no BDR, the neighboring router with the highest priority is elected the DR, and the router with the next highest priority is elected the BDR. If the DR goes off-line, the BDR automatically becomes the DR, and the router with the next highest priority then becomes the new BDR. If multiple routing switches on the same OSPF network are declaring themselves DRs, both priority and router ID are used to select the DR and BDRs.

Priority is configurable using the `ip ospf priority` command at the interface level. If two neighbors share the same priority, the router with the highest router ID is elected as the DR. The router with the next highest router ID is elected as the BDR.

OSPFv2 area types

OSPFv2 is built upon a hierarchy of network areas. All areas for a given OSPF domain reside in the same AS. An AS is defined as a number of contiguous networks that share the same interior gateway routing protocol.

An AS can be divided into multiple areas. Each area represents a collection of contiguous networks and hosts, and the topology of a given area is not known by the internal routers in any other area. Areas define the boundaries to which types 1 and 2 LSAs are broadcast, which limits the amount of LSA flooding that occurs within the AS and also helps to control the size of the LSDBs maintained in OSPF routers. An area is represented in OSPF by either an IP address or a number. Area types include: Backbone, Normal, Not-so-stubby (NSSA), and stub.

Backbone area

Every AS must have one (and only one) backbone area (identified as area 0 or 0.0.0.0.) The ABRs of all other areas in the same AS connect to the backbone area, either physically through an ABR or through a

configured virtual link. The backbone is a transit area that carries the type-3 summary LSAs, type-5 AS external link LSAs and routed traffic between non-backbone areas, as well as the type-1 and type-2 LSAs and routed traffic internal to the area. ASBRs are allowed in backbone areas.

Normal area

Normal area connects to the backbone area through one or more ABRs (physically or through a virtual link) and supports type-3 summary LSAs and type-5 external link LSAs to and from the backbone area. ASBRs are allowed in normal areas.

Stub area

Stub area connects to the AS backbone through one or more ABRs. It does not allow an internal ASBR, and does not allow external (type 5) LSAs. A stub area supports these actions:

- Advertise the area summary routes to the backbone area.
- Advertise summary routes from other areas.
- Use the default summary (type-3) route to advertise both of the following:
 - Summary routes to other areas in the AS
 - External routes to other ASs

You can configure the stub area ABR to do the following:

- Suppress advertising from some or all area summarized internal routes into the backbone area.
- Suppress LSA traffic from other areas in the AS by replacing type-3 summary LSAs and the default external route from the backbone area with the default summary route (0.0.0.0/0.)

Virtual links are not allowed for stub areas.

Not-so-stubby (NSSA) area

NSSA area connects to the backbone area through one or more ABRs. NSSAs are intended for use where an ASBR exists in an area where you want to control the following:

- Advertising the ASBR external route paths to the backbone area
- Allowing LSAs from the backbone area to advertise in the NSSA:
 - Summary routes (type-3 LSAs) from other areas
 - External routes (type-5 LSAs) from other areas as a default external route (type-7 LSAs)

Virtual links are not allowed for NSSAs.

OSPFv2 configuration task list

Tasks at a glance

- **Enabling OSPF on the routing switch**
- **Assigning the routing switch to an OSPF area**
- **Setting OSPF network for the area**
- **Creating an OSPF virtual link for an area**

- [Configuring external route redistribution and control](#)
- [Configuring ranges on an ABR to reduce advertising to the backbone](#)
- [Influencing route choice by changing the administrative distance](#)
- [Configuring graceful restart of OSPF routing](#)
- [Configuring OSPF interface settings](#)
- [Configuring OSPF interface authentication](#)
- [Configuring OSPF virtual link settings](#)
- [Configuring OSPF authentication on a virtual link](#)
- [Configuring all OSPF interfaces as passive](#)
- [Viewing OSPFv2 information](#)
- [Clearing OSPF statistics on a switch](#)

Configuring OSPF on the routing switch

Create the OSPF instance and enter the OSPF router configuration context. From this, you can proceed with other OSPF configuration tasks.

Prerequisites

- You must be in the global configuration context, as indicated by the `switch(config) #` prompt to create the OSPF instance and enter the OSPF router configuration context.
- To configure a router ID, create OSPF network areas, or adjust other global OSPF configuration items, you must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

1. Create the OSPF instance and enter the OSPF router configuration context using the following command. For command details, see [router ospf](#).

```
router ospf <PROCESS-ID> [vrf <VRF-NAME>]
```

For example, the following command creates OSPF instance 1.

```
switch(config) # router ospf 1
switch(config-ospf-1) #
```

2. Configure a global router ID using the following command. For command details, see [router-id](#).

```
router-id <ROUTER-ADDRESS>
```

For example, the following command sets the router ID to 1.1.1.1.

```
switch(config-ospf-1) # router-id 1.1.1.1
```

3. Optionally, if the OSPF process was disabled (it is enabled by default), enable the OSPF process using the following command.

```
enable
```

For command details, see [enable](#). (Refer to [disable](#) for disabling the OSPF process).

Assigning the routing switch to an OSPF area

Create a Normal, Stub, or Not So Stubby (NSSA) area.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

1. Create an OSPF area for the routing switch using one of the following commands:

- Create a normal area using the following command: `area <area-id>`. For command details, see [area](#).
- Create a stub area using the following command: `area <area-id> stub`. For command details, see [area stub](#).
- Create a Not-So-Stubby-Area using the command: `area <area-id> nssa`. For command details, see [area nssa](#).

For example, the following command creates a normal area with an area identifier of 10.1.1.1. Area identifier could alternatively be entered in decimal format such as 1.

```
switch(config-ospf-1) # area 10.1.1.1
```

Setting OSPF network for the area

After you define an OSPF area, you can assign one or more networks to it. The OSPF protocol will run on the interface with the configured IPv4 address. The interfaces which have an IP address configured in this network or in a subset of this network, will participate in the OSPF protocol.

Prerequisites

You must be in the interface configuration context, as indicated by the `switch(config-if) #` prompt.

Procedure

1. Set an OSPF network for the area using the following command. For command details, see [ip ospf area](#).

```
ip ospf <PROCESS-ID> area <AREA-ID>
```

For example, use the following command to assign interface 1/1/1 to OSPF area 1. The area can alternatively be entered as an IPv4 address.

```
switch(config) # interface 1/1/1  
switch(config-if) # ip ospf 1 area 1
```

2. Optionally, you can disable OSPF on the interface using the following command. For command details, see [ip ospf shutdown](#).

```
ip ospf shutdown
```

```
switch(config) # interface 1/1/1  
switch(config-if) # ip ospf shutdown
```

Creating an OSPF virtual link for an area

Create an OSPF virtual link with remote ABR (if not created already) and enter the vlink context.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

Create an OSPF virtual link using the following command. For command details, see [area virtual-link](#).

```
area <area-id> virtual-link <router-id>
```

For example, the following command creates a virtual link in area 100.

```
switch(config-ospf-1) # area 100 virtual-link 100.0.1.1
switch(config-router-vlink) #
```

Configuring external route redistribution and control

Configuring route redistribution for OSPF establishes the routing switch as an ASBR for importing and translating different protocol routes into OSPF. When you configure redistribution for OSPF, you can specify that static, connected, or BGP routes external to the OSPF domain are imported as OSPF routes.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

1. Enable route redistribution using the command [redistribute](#).

```
redistribute {bgp | connected | static | rip | ospf <PROCESS-ID>} [route-map <ROUTE-MAP-NAME>]
```

For example, the following command sets redistribution of connected routes as OSPF routes:

```
switch(config-ospf-1) # redistribute connected
```

2. Optionally, modify the default metric for redistribution using the following command. For command details, see [default-metric](#).

```
default-metric <METRIC-VALUE>
```

For example, the following command sets the default metric for redistribution to 37.

```
switch(config-ospf-1) # default-metric 37
```

3. Optionally, set the cost of default-summary LSAs using the following command. For command details, see [area default-metric](#).

```
area <AREA-ID> default-metric <COST>
```

For example, the following command sets the cost of default summary LSAs to 2.

```
switch (config-ospf-1) # area 1 default-metric 2
```

4. Optionally, set the protocol to advertise a maximum metric so that other routers do not prefer this router as an intermediate hop in their shortest path first (SPF) calculations. For command details, see [max-metric router-lsa](#).

```
max-metric router-lsa [on-startup [<ADVERT-TIME>]]
```

For example, the following command sets advertise max-metric router-lsa on startup.

```
switch(config-ospf-1) # max-metric router-lsa on-startup 3000
```

5. Set the maximum number of ECMP routes that OSPF can support using the following command. For command details, see [maximum-paths](#).

```
maximum-paths <MAX-VALUE>
```

For example, the following command sets the maximum number of ECMP routes to 8.

```
switch(config-ospf-1) # maximum-paths 8
```

Configuring area ranges on an ABR to reduce advertisements to the backbone

You can configure area ranges to reduce inter-area advertisements by summarizing a range of IP addresses into a single route advertisement. This action prevents an ABR from advertising specific networks or subnets to the backbone area.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

Summarize inter-area or NSSA paths using the following command. For command details, see [area range](#).

```
area <area-id> range <ip-prefix> type {inter-area | nssa} [no-advertise]
```

For example, use the following command to summarize routes matching the area range 172.77.114.0/24 using inter-area as the type of address aggregation.

```
switch(config-ospf-1) # area 1 range 172.77.114.0/24 type inter-area
```

In another example, use the following command to specify DoNotAdvertise status for routes matching the area range 172.77.114.0/24. Use nssa as the type of address aggregation.

```
switch(config-ospf-1) # area 1 range 172.77.114.0/24 type nssa no-advertise
```

Influencing route choice by changing the administrative distance

The administrative distance value can be left in its default configuration setting (110) unless a change is needed to improve OSPF performance for a specific network configuration.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

Reconfigure the administrative distance using the following command. For command details, see [distance](#) command.

`distance <distance>`

For example, use the following command to set administrative distance to 100.

```
switch(config-ospf-1) # distance 100
```

Configuring graceful restart of OSPF routing

OSPF routing can be gracefully restarted on the switch without losing packets that are in transit. OSPF neighbors are informed that the router is completing a graceful restart, which allows for maintenance on the switch without interrupting traffic in the network. There is no effect on the saved switch configuration.

Prerequisites



NOTE: Graceful restart is only applicable to the 6200-VSF, 6300-VSF, 6400, and 8400 switches.

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

Enable graceful restart of OSPF routing using the following command. For command details, see [graceful-restart](#).

```
graceful-restart {restart-interval <seconds> | helper}
```

For example, the following command specifies 50 seconds as the maximum interval another router will wait for this router to gracefully restart.

```
switch(config-ospf-1) # graceful-restart restart-interval 50
```

Configuring OSPF interface settings

You can optionally adjust the following OSPF interface settings.

Prerequisites

You must be in the interface configuration context, as indicated by the `switch(config-if) #` prompt.

Procedure

- Set the interface cost using the following command. For command details, see [ip ospf cost](#).

```
ip ospf cost <interface-cost>
```

For example, the following command sets the cost associated with interface 1/1/1 to 100.

```
switch(config) # interface 1/1/1
switch(config-if) # ip ospf cost 100
```

- Set the time interval between OSPF hello packets for the OSPF interface using the following command. For command details, see [ip ospf hello-interval](#).

```
ip ospf hello-interval <seconds>
```

- Set the interval after which a neighbor is declared dead if no hello packet is received on the OSPF interface using the following command. For command details, see [ip ospf dead-interval](#).

```
ip ospf dead-interval <seconds>
```


- Set the time between retransmitting lost link state advertisements for the OSPF interface using the following command. For command details, see [ip ospf retransmit-interval](#).
`ip ospf retransmit-interval <seconds>`
- Sets the transit delay in link state transmission for the OSPF interface using the following command. For command details, see [ip ospf transit-delay](#).
`ip ospf transit-delay <seconds>`
- Set the OSPF network type for the interface. For command details, see [ip ospf network](#).
`ip ospf network {broadcast|point-to-point}`
- Set the OSPF priority on the interface using the following command. For command details, see [ip ospf priority](#).
`ip ospf priority <number-value>`
- Set the OSPF interface as OSPF passive interface. The interface participates in OSPF but does not send or receive packets on that interface. For command details, see [ip ospf passive](#).
`ip ospf passive`

Configuring OSPF interface authentication

Configure authentication on the interface. Only one method of authentication can be active on an interface at a time. If one method is already configured on an interface, configuring an alternative method on the same interface automatically overwrites the first method used.

Prerequisites

You must be in the interface configuration context, as indicated by the `switch(config-if) #` prompt.

Procedure

1. Set the authentication type that will be used for authentication with the neighbor router using the following command. For command details, see .

```
ip ospf authentication {message-digest | simple-text | null | keychain}
```

For example, the following command sets the authentication type to simple-text.

```
switch(config-if) # ip ospf authentication simple-text
```

2. For simple-text authentication, set the password using the following command. For command details, see [ip ospf authentication-key](#).

```
ip ospf authentication-key <PASSWORD>
```

For example:

```
switch(config-if) # ip ospf authentication-key secure
```

3. For MD5 authentication, set the password using the following command. For command details, see [ip ospf message-digest-key md5](#).

```
ip ospf message-digest-key {Key ID} md5 {ciphertext | plaintext} <KEY>
```

For example:

```
switch(config-if) # ip ospf message-digest-key 1 md5 ciphertext 100
```

Configuring OSPF virtual link settings

The OSPF interface parameters for this process are automatically set to their default values for virtual links. You can optionally adjust the following OSPF virtual link settings.

Prerequisites

You must be in the router vlink configuration context, as indicated by the `switch(config-router-vlink) #` prompt.

Procedure

- Set the time interval between OSPF hello packets for the OSPF virtual links using the following command. For command details, see [hello-interval](#).

```
hello-interval <seconds>
```

For example:

```
switch(config) # router ospf 1
switch(config-ospf-1) # area 100 virtual-link 100.0.1.1
switch(config-ospf-1) # hello-interval 30
```

- Set the interval which a neighbor is declared dead if no hello packet is received on the OSPF virtual links. Use the following command. For command details, see [dead-interval](#).

```
dead-interval <seconds>
```

- Set the time between retransmitting lost link state advertisements for the OSPF virtual links using the following command. For command details, see [retransmit-interval](#).

```
retransmit-interval <seconds>
```

- Sets the transit delay in Link state transmission for the OSPF virtual links using the following command. For command details, see [transit-delay](#)

```
transit-delay <seconds>
```

Configuring OSPF authentication on a virtual link

Set the OSPF virtual-link authentication type that will be used for authentication with the remote ABR when the command is used in the vlink context.

OSPF supports the same methods of authentication for virtual links as it does for interfaces in an area. In the default configuration, OSPF authentication is disabled. Only one method of authentication can be active on a virtual link at a time. If one method is configured on a virtual link, configuring the alternative method on the same link automatically replaces the first method. Both ends of a virtual link must use the same authentication method and related credentials.

Prerequisites

You must be in the router vlink configuration context, as indicated by the `switch(config-router-vlink) #` prompt.

Procedure

1. Set the OSPF virtual-link authentication type using the following command. For command details, see [authentication](#).

```
authentication {message-digest | simple-text | null}
```

For example, the following command sets virtual-link authentication type to simple-text.

```
switch(config-router-vlink)# authentication simple-text
```

2. For simple-text authentication, set the password using the following command. For command details, see [authentication-key](#).

```
authentication-key <password>
```

For example:

```
switch(config-router-vlink)# authentication-key plaintext secure
```

3. For MD5 authentication, set the password using the following command. For command details, see [message-digest-key md5](#).

```
message-digest-key md5 <key>
```

For example:

```
switch(config-router-vlink)# message-digest-key 1 md5 plaintext secure
```

Configuring all OSPF interfaces as passive

OSPF sends LSAs to all other routers in the same AS. To limit the flooding of LSAs throughout the AS, you can configure OSPF to be passive.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router)#` prompt.

Procedure

Configure all OSPF interfaces as passive using the following command. For command details, see [passive-interface default](#).

```
passive-interface default
```

```
switch(config-ospf-1)# passive-interface default
```

Configuring SPF throttling timers

SPF calculation is throttled with default timer values (start-time 200ms, hold-time 1000ms, max-wait-time 5000ms). You can throttle SPF calculation by configuring non-default timers to improve performance of a specific network configuration.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router)#` prompt.

Procedure

Reconfigure SPF throttling timers using the following command. For command details, see [timers throttle spf](#)

```
timers throttle spf start-time <milliseconds> hold-time <milliseconds> max-wait-time <milliseconds>
```

For example, use the following command to set start-time to 500ms, hold-time to 3000ms and max-wait-time to 9000ms:

```
switch(config-ospf-1)# timers throttle spf start-time 500 hold-time 3000 max-wait-time 9000
```

Viewing OSPFv2 information

Prerequisites

Use these show commands from the Manager context, as indicated by the `switch#` prompt.

Procedure

1. To view OSPFv2 information, use the following commands. For command details and examples, click the links.

- To view general OSPF, area, state and configuration information use: [show ip ospf](#).
- To view OSPF routing table entries for Area Border Router (ABR) and Autonomous System Border Router (ASBR) use: [show ip ospf border-routers](#).
- To view OSPF link state database summary for different OSPF LSAs (Link State Advertisement) use: [show ip ospf lsdb](#).
Many different parameters are available with this command to display information for a particular LSA.
- To view information about OSPF-enabled interfaces use: [show ip ospf interface](#).
- To view information about OSPF neighbors use: [show ip ospf neighbors](#).
- To view OSPF routing table information use: [show ip ospf routes](#).
- To view OSPF statistics use: [show ip ospf statistics](#).
- To view OSPF statistics for the OSPF-enabled interfaces use [show ip ospf statistics interface](#).

- To view the current state and parameters of the OSPF virtual-links use **show ip ospf virtual-links**.

Clearing OSPF statistics on a switch

Procedure

Clear OSPF event statics using the following command. For command details, see **clear ip ospf statistics**.

```
clear ip ospf [<PROCESS-ID>] statistics [interface [<INTERFACE-NAME>]] [all-vrfs | vrf <VRF-NAME>]
```

For example, the following command clears OSPF event statistics from interface 1/1/1.

```
switch(config-router)# clear ip ospf statistics interface 1/1/1
```

An example of the OSPFv2 information in the show running-config command

When OSPFv2 is configured, the output of the `show running-config` command includes OSPFv2 information.

Example

```
switch# show running-config
Current configuration:
!
!Version ArubaOS-CX 10.0X.XXXX
!
lldp enable
timezone set utc
vrf red
mgmt
led base-loc_fdc on
led base-loc_on
led base-hlth_fdc fast_blink
led base-pwr_fdc on
!
!
!
!
!
!
aaa authentication login default local
aaa authorization commands default none
!
!
!
router ospf 1 vrf vrf_default
    area 0.0.0.0
    area 0.0.0.1
    area 0.0.0.1 virtual-link 55.4.4.4
        hello-interval 15
        dead-interval 60
```

```

router ospf 1 vrf red
  area 0.0.0.4
  area 0.0.0.5
  area 0.0.0.5 range 10.1.0.0/16 type inter-area
vlan 1
  no shutdown
interface lag 44
  no shutdown
  ip address 33.1.1.1/24
  ip ospf 1 area 0.0.0.0
  ip ospf hello-interval 22
interface 1/1/1
  no shutdown
  ip address 33.44.1.1/24
  ip ospf 1 area 0.0.0.0
  ip ospf passive
  ip ospf dead-interval 88
  ip ospf transit-delay 44
  ip ospf priority 88
  ip ospf cost 8
  ip ospf network point-to-point
  ip ospf authentication simple-text
  ip ospf authentication-key kkk
  ip ospf shutdown
interface 1/1/2
interface loopback 2
  ip address 55.55.55.55/32
  ip ospf 1 area 0.0.0.0

```

OSPFv2 commands

area

Syntax

```
area <AREA-ID>
```

```
no area <AREA-ID>
```

Description

Creates a normal area, with <AREA-ID> set if not present. If the area is already present and it is not a normal area, then this command changes the area type to normal.

The `no` form of this command deletes the area with the <AREA-ID> specified. Area can be of any type (nssa, nssa no-summary, stub, stub no-summary, and default normal area).

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating a normal area:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 1
switch(config-ospf-1)# area 10.1.1.1
Switch(config-ospf-1)# show running-config current-context router ospf 1
    router-id 1.1.1.1
    area 0.0.0.0
    area 0.0.0.1
    area 0.0.0.2 stub
    area 0.0.0.3 nssa
```

Deleting an area:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no area 1
```

area default-metric

Syntax

```
area <AREA-ID> default-metric <COST>
```

```
no area <AREA-ID> default-metric
```

Description

Sets the cost of the default route announced to NSSA or stub areas.

The `no` form of this command resets the cost of the default route announced to NSSA or stub areas, to the default value of 1.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies area ID in one of the following formats.

- OSPF area identifier in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

default-metric <COST>

Sets the cost of default-summary LSAs announced to NSSA or stub areas, to the specified value. Default cost: 1. Range: 0 to 16777215.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting cost for default LSA summary:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# area 1 default-metric 2  
switch(config-ospf-1)# area 0.0.0.1 default-metric 2
```

Setting cost for default LSA summary to default:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no area 1 default-metric  
switch(config-ospf-1)# no area 0.0.0.1 default-metric
```

area nssa

Syntax

```
area <AREA-ID> nssa [no-summary]
```

```
no area <AREA-ID> nssa [no-summary]
```

Description

Creates the NSSA area (Not So Stubby Area) with <AREA-ID> if not present. If area is present and not NSSA area, this command changes the area type to NSSA area. If `no-summary` is used, area type will be NSSA No-Summary.

The `no` form of this command unsets the area type as NSSA. That is, the configured area will be changed to default normal area. The `no area <AREA-ID> nssa no-summary` command enables sending inter-area routes into NSSA, but will not unset the area as NSSA.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

nssa [no-summary]

Specifies Not So Stubby Area (NSSA) area type. If area is present and not NSSA area, parameter changes the area type to NSSA area. If `no-summary` is specified, area type will be NSSA No-Summary, which means do not inject inter-area routes into NSSA.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating an NSSA area:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# area 1 nssa  
switch(config-ospf-1)# area 1 nssa no-summary
```

Unsetting the area as NSSA


```
switch(config)# router ospf 1
switch(config-ospf-1)# no area 1 nssa
switch(config-ospf-1)# no area 1 nssa no-summary
```

area range

Syntax

```
area <AREA-ID> range <IP-PREFIX> type {inter-area | nssa} [no-advertise]
```

```
no area <AREA-ID> range <IP-PREFIX> type {inter-area | nssa} [no-advertise]
```

Description

Summarizes the routes with the matching address or masks. This command only works for border routers.

The `no` form of this command removes route summarization for the configured IPv4 prefix address on the ABR. When using the `no` form of the command with the `no-advertise` option, enables advertising this range to other areas.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

range <IP-PREFIX>

Specifies summarizing routes matching the area range prefix/mask.

type {inter-area | nssa}

Specifies the type this address aggregation applies to as either inter-area range prefix or NSSA range prefix.

no-advertise

Specifies the address range status as `DoNotAdvertise` (do not advertise this range to other areas).

Authority

Administrators or local user group members with execution rights for this command.

Examples

Summarizing inter-area or NSSA paths:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 1
switch(config-ospf-1)# area 2 nssa
switch(config-ospf-1)# area 1 range 192.77.114.0/24 type inter-area
switch(config-ospf-1)# area 2 range 192.77.114.0/24 type nssa
switch(config-ospf-1)# area 2 range 192.77.114.0/24 type nssa no-advertise
```

Removing summarization:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no area 1 range 192.77.114.0/24 type inter-area
switch(config-ospf-1)# no area 2 range 192.77.114.0/24 type nssa
switch(config-ospf-1)# no area 2 range 192.77.114.0/24 type nssa no-advertise
```

area stub

Syntax

```
area <AREA-ID> stub [no-summary]
```

```
no area <AREA-ID> stub [no-summary]
```

Description

Creates the stub area with <AREA-ID> if not present. If the area is already present and it is not a normal stub area, then this command changes the stub area type to normal. If the `no-summary` parameter is used, area type will be stub No-Summary.

The `no` form of this command unsets the area as a stub type. That is, the configured area will be changed to a default normal area. The `no area <AREA-ID> stub no_summary` command enables sending inter-area routes into the stub area, but will not unset the area as stub.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- OSPF area identifier in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

stub [no-summary]

Specifies the stub area type. If the area is already present and it is not a stub area, this parameter changes the area type to stub. If `no-summary` is specified, area type will be stub No-Summary (totally stubby area), which means do not inject summary link advertisements into stub areas.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating a STUB area:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 1 stub
switch(config-ospf-1)# area 1 stub no-summary
```

Unsetting the area type as stub:

```
switch(config)# router ospf 1
switch(config-ospf-1) # no area 1 stub
switch(config-ospf-1) # no area 1 sub no-summary
```

area virtual-link

Syntax

```
area <AREA-ID> virtual-link <ROUTER-ID>
```

```
no area <AREA-ID> virtual-link <ROUTER-ID>
```

Description

Creates an OSPF virtual link with a remote ABR and enters the vlink context.

The `no` form of this command deletes an OSPF virtual link with the specified router ID of the remote ABR. If `no <ROUTER-ID>` is specified, the `no` form of the command sets the virtual link to the default settings.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

virtual-link <ROUTER-ID>

Configures a virtual link with the specified router ID of the remote ABR.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring OSPF virtual links:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)#
```

Deleting OSPF virtual links:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no area 100 virtual-link 100.0.1.1
```

authentication

Syntax

```
authentication {message-digest | simple-text | null | keychain}
```

```
no authentication
```

Description

Sets the OSPF virtual-link authentication type that will be used for authentication with the remote ABR.

The **no** form of this command unconfigures the virtual-link authentication type used and sets it to Null authentication.

Command context

config-router-vlink

Parameters

Choose one of the authentication types from the following parameters.

message-digest

Sets the authentication type to message-digest.

simple-text

Sets the authentication type to simple-text.

null

Sets the authentication type to null.

keychain

Sets authentication type to use the key chain.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF virtual links authentication type:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# authentication simple-text
```

Deleting OSPF virtual links authentication type:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no authentication
```

authentication-key

Syntax

```
authentication-key {ciphertext | plaintext} <PASSWORD>
```

```
no authentication-key
```

Description

Sets the OSPF virtual-link authentication password that is used for simple-text authentication. If the password is given in ciphertext, it will be decrypted and applied to the protocol.

The **no** form of this command deletes the virtual-link authentication password that is used for simple-text authentication.

Command context

config-router-vlink

Parameters

ciphertext

Specifies providing the password in encrypted format.

plaintext

Specifies providing the password in simple-text format.

<PASSWORD>

Specifies the password for simple-text authentication.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF virtual links simple-text authentication password:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# authentication-key plaintext secure
switch(config-router-vlink)# authentication-key ciphertext AQBapdAVzEZb205RmH33+4cEctMayhvmuOVeYmXs8Y7ANu2eBgAAABO5xecopg==
```

Deleting OSPF virtual links simple-text authentication password:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no authentication-key
```

clear ip ospf neighbors

Syntax

```
clear ip ospf [<PROCESS-ID>] neighbor [<NEIGHBOR>] [interface [<INTERFACE-NAME>]]
[all-vrfs | vrf <VRF-NAME>]
```

Description

Resets the neighbor and clears the OSPF neighbor information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies the OSPFv2 process ID to clear the statistics for the particular OSPFv2 process. Range: 1 to 63.

<NEIGHBOR>

Specifies the router ID of a neighbor.

<INTERFACE-NAME>

Specifies the OSPFv2 statistics to clear for the specified interface.

all-vrfs

Select to clear the OSPFv2 statistics for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Example

Clearing the OSPFv2 neighbor information:

```
switch# clear ipv6 ospfv3 1 neighbor
switch# clear ip ospf 1 neighbor 3.3.3.3
switch# clear ip ospf 1 neighbor interface 1/1/1
switch# clear ip ospf neighbor 5.5.5.5 vrf red
```

clear ip ospf statistics

Syntax

```
clear ip ospf [<PROCESS-ID>] statistics [interface [<INTERFACE-NAME>]] [all-vrfs | vrf <VRF-NAME>]
```

Description

Clear the OSPF event statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

OSPF process ID. Clear the statistics for the particular OSPF process. Range: 1 to 63.

<INTERFACE-NAME>

Clear the OSPF statistics for the specified interface.

all-vrfs

Optionally select to clear the OSPF statistics for all VRFs.

vrf <VRF-NAME>

Optionally select to clear the OSPF statistics for a particular VRF. If the VRF is not specified, information for the default VRF is cleared.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

Clearing the OSPF event statistics:

```
switch# clear ip ospf statistics
switch# clear ip ospf statistics interface 1/1/1
```

dead-interval

Syntax

```
dead-interval <INTERVAL>
```

```
no dead-interval
```

Description

Sets the interval after which a neighbor is declared dead if no hello packet comes in for virtual links.

The `no` form of this command sets the dead interval to default for virtual links. The default value is 40 seconds (generally four times the hello packet interval).



NOTE: For proper operation, set the dead interval must be longer than the hello interval.

Command context

```
config-router-vlink
```

Parameters

<INTERVAL>

Specifies the time interval for the dead interval, in seconds. Range: 1 to 65535. Default: 40.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the OSPv2F virtual links dead interval:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# dead-interval 30
```

Setting the OSPFv2 virtual links dead interval to default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no dead-interval
```

default-information originate

Syntax

```
default-information originate
```

```
no default-information originate
```

Description

Configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors if it is present in the routing table.

The `no` form of this command disables advertisement of the default route.

Command context

config-ospf-<PROCESS-ID>

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# default-information originate
```

Disabling advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no default-information originate
```

default-information originate always

Syntax

default-information originate always

no default-information originate always

Description

Configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors, regardless if it is present in the routing table or not.

The **no** form of this command sets the OSPF administrative distance to the default of 110.

Command context

config-ospf-<PROCESS-ID>

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# default-information originate always
```

Disabling advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no default-information originate always
```

default-metric

Syntax

default-metric <METRIC-VALUE>

no default-metric

Description

Sets the default metric for redistributed routes in the OSPF.

The `no` form of this command sets the default metric to be used for redistributed routes into OSPF to the default of 25.

Command context

`config-ospf-<PROCESS-ID>`

Parameters

<METRIC-VALUE>

Specifies the default metric value to use for redistributed routes. Default: 25. Range: 0-1677214.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting default metric for redistributed routes:

```
switch(config)# router ospf 1
switch(config-ospf-1)# default-metric 37
```

Setting default metric for redistributed routes to the default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no default-metric
```

disable

Syntax

`disable`

Description

Disables the OSPF process.



IMPORTANT: This command does not remove the OSPF configurations.

Command context

`config-ospf-<PROCESS-ID>`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling OSPF process:

```
switch(config)# router ospf 1
switch(config-ospf-1)# disable
```

distance

Syntax

```
distance <DISTANCE>
```

```
no distance
```

Description

Defines an administrative distance for OSPF. Administrative distance is used as a criteria to select the best route when multiple routes are present from different routing protocols.

The `no` form of this command sets the OSPF administrative distance to the default of 110.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<DISTANCE>

Specifies OSPF administrative distance. Range: 1-255. Default: 110.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF administrative distance:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# distance 100
```

Setting OSPF administrative distance to the default:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no distance
```

enable

Syntax

```
enable
```

Description

Enables the OSPF process, if disabled. By default the OSPF process is enabled.

Command context

```
config-ospf-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling OSPF process:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# enable
```

graceful-restart

Syntax

```
graceful-restart {restart-interval <SECONDS> | helper}
```

```
no graceful-restart {restart-interval | helper}
```

Description

Configures graceful restart parameters for OSPF.

The `no` form of this command sets the restart interval to the default interval of 120 seconds or disables the helper mode, depending on the parameter specified.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

Choose one of the following parameters:

restart-interval <SECONDS>

Specifies the time another router should wait for this router to gracefully restart and selects the maximum interval (in seconds) that another router should wait. Default: 120 seconds. Range: 5-1800.

helper

Specifies that the router will participate in the graceful restart of a neighbor router.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling OSPF nonstop forwarding:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# graceful-restart restart-interval 50  
switch(config-ospf-1)# graceful-restart helper
```

Setting restart-interval to default, disable helper mode:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no graceful-restart restart-interval  
switch(config-ospf-1)# no graceful-restart helper
```

hello-interval

Syntax

```
hello-interval <INTERVAL>
```

```
no hello-interval
```

Description

Sets the time interval between OSPF hello packets for virtual links.

The `no` form of this command sets the hello interval to the default value of 10 seconds for virtual links.



NOTE: For proper operation, the hello interval must be shorter than the dead interval.

Command context

config-router-vlink

Parameters

<INTERVAL>

Specifies the time interval for the hello interval, in seconds. Range: 1 to 65535. Default: 10.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the OSPF virtual links hello interval:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# hello-interval 30
```

Setting the OSPF virtual links hello interval to default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no hello-interval
```

ip ospf area

Syntax

```
ip ospf <PROCESS-ID> area <AREA-ID>
```

```
no ip ospf <PROCESS-ID> area <AREA-ID>
```

Description

Runs the OSPF protocol on the interface with the configured IPv4 address for the area specified. The interfaces which have an IP address configured in this network or in a subset of this network, will participate in the OSPF protocol.

To move an interface to a new area, unmap the existing area and then associate a new area with the interface.

The `no` form of this command disables OSPF on the interface and removes the interface from the area. Interfaces which have an IP address configured on the network or in a subset of the network, stop participating in the OSPF protocol.

Command context

config-if

config-if-vlan

Parameters

<PROCESS-ID>

Specifies the OSPF process Id. Range: 1 to 63.

<AREA-ID>

Specifies the OSPF area ID in one of the following formats.

- Area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- Area identifier in decimal format. Range: 0 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF network for the area:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf 1 area 1
switch(config-if-vlan)# ip ospf 1 area 0.0.0.1
```

Disabling OSPF network for the area:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf 1 area 1
switch(config-if-vlan)# no ip ospf 1 area 0.0.0.1
```

ip ospf authentication

Syntax

```
ip ospf authentication {message-digest | simple-text | null | keychain}
```

```
no ip ospf authentication
```

Description

Sets the authentication type that will be used for authentication with the neighbor router.

The `no` form of this command deletes the authentication type used for a particular authentication with the neighbor router and sets to null authentication.

Command context

config-if

config-if-vlan

Parameters

Choose one of the following authentication types.

message-digest

Sets authentication type as message-digest.

simple-text

Sets authentication type as simple-text.

null

Sets authentication type as null.

keychain

Sets the authentication type to use the key chain.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF authentication type on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf authentication simple-text
```

Deleting OSPF authentication type on the interface and sets it to null:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf authentication
```

ip ospf authentication-key

Syntax

```
ip ospf authentication-key {ciphertext | plaintext} <PASSWORD>
```

```
no ip ospf authentication-key
```

Description

Sets the authentication password used for simple-text authentication. If the password is given in ciphertext it will be decrypted and applied to the protocol.

The **no** form of this command deletes the authentication password used for simple-text authentication.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

ciphertext

Specifies providing the password in encrypted format.

plaintext

Specifies providing the password in simple-text format.

<PASSWORD>

Specifies the password for simple-text authentication.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF simple-text authentication password on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf authentication-key ciphertext AQBapdAVzEZb205RmH33+4cEctMayhvmuOVeYmXs8Y7ANu2eBgAAAB05xecopg==
switch(config-if-vlan)# ip ospf authentication-key plaintext secure
```

Deleting OSPF simple-text authentication password on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf authentication-key
```

ip ospf cost

Syntax

```
ip ospf cost <INTERFACE-COST>
```

```
no ip ospf cost
```

Description

Sets the cost (metric) associated with a particular interface. The interface cost is used as a parameter to calculate the best routes.

The `no` form of this command sets the cost (metric) associated with a particular interface to the default cost 1.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<INTERFACE-COST>

Specifies the interface cost value. Range: 1 to 65535. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF interface cost

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf cost 100
```

Setting the OSPF interface cost to default

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf cost
```

ip ospf dead-interval

Syntax

```
ip ospf dead-interval <INTERVAL>
```

```
no ip ospf dead-interval
```

Description

Sets the interval after which a neighbor is declared dead if no hello packet is received on the OSPF interface.

The **no** form of this command sets the interval after which a neighbor is declared dead, to the default for the OSPF interface. The default value is 40 seconds (generally 4 times the hello packet interval).

Command context

```
config-if  
config-if-vlan
```

Parameters

<INTERVAL>

Specifies the time interval for the dead interval, in seconds. Range: 1 to 65535. Default: 40.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF dead interval on the interface:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ip ospf dead-interval 30
```

Setting OSPF dead interval to default on the interface:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ip ospf dead-interval
```

ip ospf hello-interval

Syntax

```
ip ospf hello-interval <INTERVAL>  
  
no ip ospf hello-interval
```

Description

Sets the time interval between OSPF hello packets for the OSPF interface.

The **no** form of this command sets the time interval OSPF hello packets to the default of 10 seconds for the OSPF interface.

Command context

```
config-if  
config-if-vlan
```

Parameters

<INTERVAL>

Specifies the time interval for the hello interval, in seconds. Range: 1 to 65535. Default: 10.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF hello interval on the interface:


```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf hello-interval 30
```

Setting OSPF hello interval to the default on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf hello-interval
```

ip ospf keychain

Syntax

```
ip ospf keychain <KEYCHAIN-NAME>
```

```
no ip ospf keychain
```

Description

Sets the key chain for md5 authentication. A key chain configures rotating keys for packet authenticating, reducing the risk of keys being compromised.

The `no` form of this command deletes the key chain used for md5 authentication.

Command context

config-if

Parameters

<KEYCHAIN-NAME>

Name of key chain to be used for md5 authentication.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv2 key chain authentication:

```
switch(config)# interface 1/1/1
switch(config-if)# ip ospf keychain ospf_keys
```

Deleting OSPFv2 key chain authentication:

```
switch(config)# interface 1/1/1
switch(config-if)# no ip ospf keychain
```

ip ospf message-digest-key md5

Syntax

```
ip ospf authentication message-digest
```

```
ip ospf message-digest-key <KEY-ID> md5 {ciphertext | plaintext} <KEY>
```

```
no ip ospf message-digest-key <KEY-ID>
```

Description

Sets the message digest password used for message digest authentication. If the md5 key is given in ciphertext, it will be decrypted and applied to the protocol.

The `no` form of this command deletes the message digest password used for message digest authentication.

Command context

```
config-if
config-if-vlan
```

Parameters

ciphertext

Specifies providing the md5 key in encrypted format.

plaintext

Specifies providing the md5 key in simple-text format.

<KEY-ID>

Specifies the password for md5 (message-digest) authentication. Range: 1 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF message-digest authentication password on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf message-digest-key 1 md5 ciphertext AQBapdAVzEZb205RmH33+4cEctMayhvmuOveYmXs8Y7ANu2eBgAAAB05xecopg==
switch(config-if-vlan)# ip ospf message-digest-key 1 md5 plaintext secure
```

Deleting OSPF message-digest authentication password on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf message-digest-key 1
```

ip ospf network

Syntax

```
ip ospf network {broadcast | point-to-point}

no ip ospf network
```

Description

Configures the OSPF network type for the interface.

The `no` form of this command sets the network type for the interface to the system default which is broadcast network.

Command context

```
config-if
config-if-vlan
```

Parameters

Choose one of the following parameters as the interface network type.

broadcast

Specifies the OSPF network type as a broadcast multi-access network.

point-to-point

Specifies the OSPF network type as a point-to-point network.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF network type for the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf network broadcast
switch(config-if-vlan)# ip ospf network point-to-point
```

Disabling OSPF network type for the interface to system default of broadcast network:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf network
```

ip ospf passive

Syntax

```
ip ospf passive
```

```
no ip ospf passive
```

Description

Configures the interface as an OSPF passive interface. With this setting, the interface participates in OSPF but does not send or receive packets on that interface.

The `no` form of this command resets the interface as active. With this setting, the interface starts sending and receiving OSPF packets.

Command context

```
config-if
```

```
config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the interface as OSPF passive interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf passive
```

Setting the interface as OSPF active interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf passive
```

ip ospf priority

Syntax

```
ip ospf priority <PRIORITY-VALUE>
```

```
no ip ospf priority
```

Description

Sets the OSPF priority for the interface. The larger the numeric value of the priority, the higher the chances for it to become the designated router. Setting a priority of zero makes the router ineligible to become a designated router or back up designated router.

The `no` form of this command sets the OSPF priority for the interface to the default of 1.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<PRIORITY-VALUE>

Specifies the OSPF priority value. Range: 0 to 255. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF priority for the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf priority 50
```

Disabling OSPF priority for the interface to default:

```
switch(config)# interface vlan1
switch(config-if-vlan)# no ip ospf priority
```

ip ospf retransmit-interval

Syntax

```
ip ospf retransmit-interval <INTERVAL>
```

```
no ip ospf retransmit-interval
```

Description

Sets the time between retransmitting lost link state advertisements for the OSPF interface.

The `no` form of this command sets the time between retransmitting lost link state advertisements to the default of 5 seconds for the OSPF interface.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<INTERVAL>

Specifies the retransmit interval, in seconds. Range: 1 to 3600. Default: 5.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF retransmit interval on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf retransmit-interval 30
```

Setting OSPF retransmit interval to the default on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf retransmit-interval
```

ip ospf shutdown

Syntax

```
ip ospf shutdown
```

```
no ip ospf shutdown
```

Description

Disables OSPF on the interface. The interface state changes to Down. It does not remove the interface from the OSPF area. To remove the interface, use the command `no ip ospf area`.

The `no` form of this command re-enables OSPF on the interface

Command context

```
config-if
```

```
config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling OSPF on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf shutdown
```

Re-enabling OSPF on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf shutdown
```

ip ospf transit-delay

Syntax

```
ip ospf transit-delay <DELAY>
```

```
no ip ospf transit-delay
```

Description

Sets the time delay in link state transmission for the OSPF interface.

The `no` form of this command sets the delay in link state transmission to the default of 1 second for the OSPF interface.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<DELAY>

Specifies the transit delay in seconds. Range: 1 to 3600. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF transit delay on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ip ospf transit-delay 30
```

Setting OSPF transit delay to the default on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ip ospf transit-delay
```

keychain

Syntax

```
keychain <KEYCHAIN-NAME>
```

```
no keychain
```

Description

Sets the key chain for md5 authentication. A key chain configures rotating keys for packet authenticating, reducing the risk of keys being compromised.

The `no` form of this command deletes the key chain used for md5 authentication.

Command context

```
config-router-vlink
```

Parameters

<KEYCHAIN-NAME>

Name of key chain to be used for md5 authentication.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF virtual link key chain authentication:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# keychain ospf_keys
```

Deleting OSPF virtual link key chain authentication:

```
switch# configure terminal
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no keychain
```

max-metric router-lsa

Syntax

```
max-metric router-lsa [on-startup [<ADVERT-TIME>]]
```

```
no max-metric router-lsa [on-startup]
```

Description

Sets the protocol to advertise a maximum metric so that other routers do not prefer this router as an intermediate hop in their shortest path first (SPF) calculations. If the on-startup parameter is used, the router is configured to advertise a maximum metric at startup for the time mentioned in seconds or for a default value of 600 seconds.

The `no` form of this command advertises the normal cost metrics instead of advertising the maximized cost metric. This setting causes the router to be considered in traffic forwarding.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

on-startup <ADVERT-TIME>

Specifies the time in seconds to advertise self as stub-router on startup. If no time is specified, the default time of 600 seconds is used. Range: 5 to 86400. Default: 600.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting to maximize the cost metrics for Router LSA:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# max-metric router-lsa  
switch(config-ospf-1)# max-metric router-lsa on-startup  
switch(config-ospf-1)# max-metric router-lsa on-startup 3000
```

Setting to advertise the normal cost metrics instead of advertising the maximized cost metric:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no max-metric router-lsa  
switch(config-ospf-1)# no max-metric router-lsa on-startup
```

maximum-paths

Syntax

```
maximum-paths <MAX-VALUE>
```

```
no maximum-paths
```

Description

Sets the maximum number of ECMP routes that OSPF can support.

The `no` form of this command sets the maximum number of ECMP routes that OSPF can support to the default value of 4.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<MAX-VALUE>

Specifies the maximum number of ECMP routes. Range: 1 to 8. Default: 4.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting maximum number of ECMP routes:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# maximum-paths 8
```

Setting maximum number of ECMP routes to the default of 4:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no maximum-paths
```

message-digest-key md5

Syntax

```
message-digest-key md5 {ciphertext | plaintext} <KEY>
```

```
no message-digest-key
```


Description

Sets the message digest password that is used for message digest authentication. If the md5 key is given in ciphertext, it will be decrypted and applied to the protocol.

The `no` form of this command deletes the message-digest password that is used for message digest authentication

Command context

`config-router-vlink`

Parameters

ciphertext

Specifies providing the md5 key in encrypted format.

plaintext

Specifies providing the md5 key in plain-text format.

<KEY>

Specifies the password for md5 (message-digest) authentication.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF virtual links message-digest authentication password:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# message-digest-key 1 md5 plaintext 100
switch(config-router-vlink)# message-digest-key 1 md5 ciphertext AQBapTELhuC2K4KFYvyRx75TgKmGag3njxzUJvu2g00e7GJ6AwAAHPD7A==
```

Deleting OSPF virtual links message-digest authentication password:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no message-digest-key {key-id 1-255}
```

passive-interface default

Syntax

`passive-interface default`

`no passive-interface`

Description

Configures all OSPF interfaces as passive.

The `no` form of this command sets all OSPF interfaces as active.

Command context

`config-ospf-<PROCESS-ID>`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF-enabled interfaces as passive:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# passive-interface default
```

Setting OSPF-enabled interfaces as active:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no passive-interface
```

redistribute

Syntax

```
redistribute {bgp | connected | static | rip | ospf <PROCESS-ID>} [route-map <ROUTE-MAP-NAME>]  
no redistribute {bgp | connected | static | rip | ospf <PROCESS-ID>} [route-map <ROUTE-MAP-NAME>]
```

Description

Redistributes routes originating from other protocols, or from another OSPF process, to the current OSPF process.

The **no** form of this command disables redistribution of routes to the current OSPF process.

Command context

config-ospf-<PROCESS-ID>

Parameters

bgp

Specifies redistributing BGP (Border Gateway Protocol) routes.

connected

Specifies redistributing connected (directly attached subnet or host).

static

Specifies redistributing static routes.

rip

Specifies redistributing RIP routes.

ospf <PROCESS-ID>

Specifies redistributing routes from the specified OSPFv2 process ID.

route-map <ROUTE-MAP-NAME>

Specifies the name of a route map. To create a route map, use the **route-map** command.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Redistributing routes to OSPF:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# redistribute connected
```

```
switch(config-ospf-1)# redistribute bgp
switch(config-ospf-1)# redistribute static
switch(config-ospf-1)# redistribute static route-map static_networks
switch(config-ospf-1)# redistribute rip
switch(config-ospf-1)# redistribute ospf 2
```

Disabling redistributing routes to OSPF:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no redistribute connected
switch(config-ospf-1)# no redistribute bgp
switch(config-ospf-1)# no redistribute static
switch(config-ospf-1)# no redistribute rip
switch(config-ospf-1)# no redistribute ospf 2
```

reference-bandwidth

Syntax

```
reference-bandwidth <BANDWIDTH>
```

```
no reference-bandwidth
```

Description

Sets the reference bandwidth for OSPFv2. If the OSPFv2 interface cost is not explicitly set, then the cost of all the OSPFv2 interfaces is recalculated based on the reference bandwidth and link speed of the interface.

For VLAN interfaces the link speed value is taken as 1 Gbps (if the OSPFv2 interface cost is not explicitly set).

The `no` form of this command sets the reference bandwidth for OSPF to the default of 100000 Mbps.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<BANDWIDTH>

Specifies the reference bandwidth used to calculate the cost of an interface in Mbps. Range: 1 to 4000000. Default: 100000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the reference bandwidth:

```
switch(config)# router ospf 1
switch(config-ospf-1)# reference-bandwidth 40000
```

Setting the reference bandwidth to the default value:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no reference-bandwidth
```

retransmit-interval

Syntax

```
retransmit-interval <INTERVAL>
```

```
no retransmit-interval
```

Description

Sets the time between retransmitting lost link state advertisements for virtual links.

The `no` form of this command sets the time between retransmitting lost link state advertisements to the default of 5 seconds for virtual links.

Command context

```
config-router-vlink
```

Parameters

<INTERVAL>

Specifies the retransmit interval in seconds. Range: 1 to 3600. Default: 5.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv2 virtual links retransmit interval:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# retransmit-interval 30
```

Setting OSPFv2 virtual links retransmit interval to default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no retransmit-interval
```

rfc1583-compatibility

Syntax

```
rfc1583-compatibility
```

```
no rfc1583-compatibility
```

Description

Enables OSPF compatibility with RFC1583 (backward compatibility). If RFC1583 compatibility is enabled, then the route cost calculation follows a different method.

The `no` form of this command disables OSPF compatibility with RFC1583 (backward compatibility). By default the RFC1583 compatibility is disabled.

Command context

```
config-ospf-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling OSPF RFC1583 compatibility:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# rfc1583-compatibility
```

Disabling OSPF RFC1583 compatibility:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no rfc1583-compatibility
```

router ospf

Syntax

```
router ospf <PROCESS-ID> [vrf <VRF-NAME>]
```

```
no router ospf <PROCESS-ID> [vrf <VRF-NAME>]
```

Description

Creates an OSPF process (if not created already) on a VRF, and switches to the OSPF router instance context. Up to eight OSPF processes are supported per VRF.

The `no` form of this command removes the OSPF instance.

Command context

config

Parameters

<PROCESS-ID>

Specifies an OSPF process ID. Range: 1 to 63.

vrf <VRF-NAME>

Specifies a VRF name for the OSPF process. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# router ospf 1  
switch(config-ospf-1)#
```

```
switch(config)# router ospf 1 vrf vrf_red
```

```
switch(config)# no router ospf 1
```

router-id

Syntax

```
router-id <ROUTER-ADDR>
```

```
no router-id
```

Description

Sets an ID for the router in an IPv4 address format.

The `no` form of this command unconfigures the router-id for the instance and sets the router-id to the default as follows: the router-id is selected dynamically as equal to the highest loopback address on the router, or the highest active interface if there are no loopback addresses. If no IP address is configured on any interfaces on the router, OSPF will not form an adjacency.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<ROUTER-ADDR>

Specifies the Router ID in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting router-id in the OSPF context:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# router-id 1.1.1.1
```

Unconfiguring router-id:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no router-id
```

show ip ospf

Syntax

```
show ip ospf [<PROCESS-ID>] [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays general OSPF, area, state, and configuration information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display general OSPF information for a particular OSPF process. Range: 1 to 63.

all-vrfs

Optionally select to display general OSPF information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing general OSPF configurations:

```
switch# show ip ospf 1
Routing Process 1 with ID : 1.1.1.1 VRF default
-----

OSPFv2 Protocol is enabled
Graceful-restart is configured
Restart Interval: 120, State: inactive
Last Graceful Restart Exit Status: none
Area Border: false
AS Border: false
SPF: Start Time: 200ms, Hold Time: 1000ms, Max Wait Time: 5000ms
LSA: Start Time: 5000ms, Hold Time: 0ms, Max Wait Time: 0ms
LSA Arrival: 1000ms
Maximum Paths to Destination: 4
Number of external LSAs 0, checksum sum 0
Summary address:
  prefix 20.1.1.0/24 advertise tag 10
Number of areas is 2, 2 normal, 0 stub, 0 NSSA
Number of active areas is 1, 1 normal, 0 stub, 0 NSSA
BFD is disabled
Reference Bandwidth: 100000 Mbps
Area (0.0.0.0) (Active)
  Interfaces in this Area: 1 Active Interfaces: 1
  Passive Interfaces: 0 Loopback Interfaces: 0
  SPF calculation has run 7 times
  Area ranges:
  Number of LSAs: 3, checksum sum 75093
Area (0.0.0.1) (Inactive)
  Interfaces in this Area: 0 Active Interfaces: 0
  Passive Interfaces: 0 Loopback Interfaces: 0
  SPF calculation has run 2 times
  Area ranges:
  ip-prefix 20.1.1.1/24, inter-area, advertise
  Number of LSAs: 0, checksum sum 0
```

show ip ospf border-routers

Syntax

```
show ip ospf [<PROCESS-ID>]
  border-routers [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays the OSPF routing table entries for Area Border Router (ABR) and Autonomous System Border Router (ASBR).

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display the OSPF routing table entries for ABR and ASBR for the particular OSPF process. Range: 1 to 63.

all-vrfs

Select to display OSPF border router information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF border routers information:

```
switch# show ip ospf border-routers
OSPF Process ID 1 VRF default Internal Routing Table
-----

Codes: i - Intra-area route, I - Inter-area route

i 2.2.2.2 [10], ABR, Area 0.0.0.0, SPF 15 via
    10.1.1.2, Interface 1/1/1
i 2.2.2.2 [10], ABR, Area 0.0.0.1, SPF 15 via
    11.1.1.12, Interface 1/1/2
```

show ip ospf interface

Syntax

```
show ip ospf [<PROCESS-ID>] interface [<interface-name>] [brief]
[all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays general OSPF, area, state, and configuration information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display general OSPF information for a particular OSPF process. Range: 1 to 63.

<interface-name>]

Specify the name of an OSPF interface.

brief

Display a brief overview of OSPF configuration information.

all-vrfs

Select to display general OSPF information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing general OSPF configuration settings for the default VRF:

```
switch# show ip ospf 1
Routing Process 1 with ID : 1.1.1.1 VRF default
-----

OSPFv2 Protocol is enabled
Graceful-restart is configured
Restart Interval: 120, State: inactive
Last Graceful Restart Exit Status: none
Area Border: false
AS Border: false
SPF: Start Time: 200ms, Hold Time: 1000ms, Max Wait Time: 5000ms
LSA: Start Time: 5000ms, Hold Time: 0ms, Max Wait Time: 0ms
LSA Arrival: 1000ms
Maximum Paths to Destination: 4
Number of external LSAs 0, checksum sum 0
Summary address:
  prefix 20.1.1.0/24 advertise tag 10
Number of areas is 2, 2 normal, 0 stub, 0 NSSA
Number of active areas is 1, 1 normal, 0 stub, 0 NSSA
BFD is disabled
Reference Bandwidth: 100000 Mbps
Area (0.0.0.0) (Active)
  Interfaces in this Area: 1 Active Interfaces: 1
  Passive Interfaces: 0 Loopback Interfaces: 0
  SPF calculation has run 7 times
  Area ranges:
  Number of LSAs: 3, checksum sum 75093
Area (0.0.0.1) (Inactive)
  Interfaces in this Area: 0 Active Interfaces: 0
```

```
Passive Interfaces: 0 Loopback Interfaces: 0
SPF calculation has run 2 times
Area ranges:
  ip-prefix 20.1.1.1/24, inter-area, advertise
Number of LSAs: 0, checksum sum 0
```

Showing OSPF configuration settings for all interfaces:

```
switch# show ip ospf interface
Interface 1/1/1 is up, line protocol is up
-----

IP address 10.1.1.1/24, Process ID 1 VRF default, area 0.0.0.0
  State Backup-dr, Status up, Network type Broadcast
  Link Speed: 10000 Mbps
  Cost Configured NA, Calculated 10
  Transit delay 1 sec, Router priority 1
  Designated Router IP: 10.1.1.2
  Backup Designated Router IP: 10.1.1.1
  Timer Intervals: Hello 10, Dead 40, Retransmit 5
  No authentication
  Number of Link LSAs: 0, checksum sum 0
  BFD is disabled

Interface 1/1/2 is up, line protocol is up
-----

IP address 11.1.1.1/24, Process ID 1 VRF default, area 0.0.0.1
  State Dr, Status up, Network type Broadcast
  Link Speed: 10000 Mbps
  Cost Configured NA, Calculated 10
  Transit delay 1 sec, Router priority 1
  Designated Router IP: 11.1.1.1
  No backup designated router on this network
  Timer Intervals: Hello 10, Dead 40, Retransmit 5
  No authentication
  Number of Link LSAs: 0, checksum sum 0
  BFD is disabled
```

show ip ospf lsdb

Syntax

```
show ip ospf [<PROCESS-ID>] lsdb [all-vrfs | vrf <VRF-NAME>]
  [area <AREA-ID>] [lsid <LINK-STATE-ID>]
  [adv-router {<ROUTER-ID> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdb [all-vrfs | vrf <VRF-NAME>]
  asbr-summary [area <AREA-ID>] [lsid <LINK-STATE-ID>]
  [adv-router {<router-id> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdb [all-vrfs | vrf <VRF-NAME>]
  external [lsid <LINK-STATE-ID>] [adv-router {<ROUTER-ID> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdb [all-vrfs | vrf <VRF-NAME>]
  router [area <AREA-ID>] [lsid <LINK-STATE-ID>]
  [adv-router {<ROUTER-ID> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdb [all-vrfs | vrf <VRF-NAME>]
  network [area <AREA-ID>] [lsid <LINK-STATE-ID>]
  [adv-router {<ROUTER-ID> | self}] [vsx-peer]
```

```
show ip ospf [<PROCESS-ID>] lsdbs [all-vrfs | vrf <VRF-NAME>]
summary [area <AREA-ID>] [lsid <LINK-STATE-ID>]
[adv-router {<ROUTER-ID> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdbs [all-vrfs | vrf <VRF-NAME>]
nssa-external [area <AREA-ID>] [lsid <LINK-STATE-ID>]
[adv-router {<ROUTER-ID> | self}] [vsx-peer]

show ip ospf [<PROCESS-ID>] lsdbs [all-vrfs | vrf <VRF-NAME>]
database-summary [vsx-peer]
```

Description

Shows the OSPF link state database summary for different OSPF LSAs (Link State Advertisement). Use the parameters to get information for a particular LSA.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display the OSPF link state database information for the specified OSPF process. Range: 1 to 63.

all-vrfs

Select to display OSPF link state database information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

Optionally select one of the following parameters to filter the link state database information.

- `asbr-summary`. Displays ASBR summary link states (LSA type 4).
- `external`. Displays external link states (LSA type 5). The external parameter does not take the area <AREA-ID> parameter.
- `router`. Displays router LSAs (LSA type 1).
- `network`. Displays network LSAs (LSA type 2).
- `summary`. Displays network-summary link states (LSA type 3).
- `nssa-external`. Displays NSSA external link states (LSA type 7).

database-summary

Select to display the count of each type of LSA and each area in the database.

area <AREA-ID>

Select to display information filtered for the specified area in one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Value: 0 to 4294967295.

lsid <LINK-STATE-ID>

Select to display information filtered by link state identifier specified in IPv4 address format (A.B.C.D).

adv-router {<ROUTER-ID> | **self**}

Select to display link states for a particular advertising router. Specify either a Router ID of the advertising router or specify **self** to show self-originated link states.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF link state database (LSDB) general information:

```
switch# show ip ospf lsdb
OSPF Router with ID (50.50.50.50) (Process ID 1 VRF default)
=====

Router Link State Advertisements (Area 0.0.0.0)
-----
```

LSID	ADV Router	Age	Seq#	Checksum	Link Count
40.40.40.40	40.40.40.40	930	0x80000004	0x2ea1	3
50.50.50.50	50.50.50.50	935	0x80000002	0x8b52	1
60.60.60.60	60.60.60.60	943	0x800003c5	0x9854	2

```
Network Link State Advertisements (Area 0.0.0.0)
-----
```

LSID	ADV Router	Age	Seq#	Checksum
209.165.201.3	60.60.60.60	944	0x80000001	0x7179
192.0.2.1	50.50.50.50	935	0x80000001	0x516a

```
Inter-area Summary Link State Advertisements (Area 0.0.0.0)
-----
```

LSID	ADV Router	Age	Seq#	Checksum
209.165.201.1	40.40.40.40	929	0x80000001	0x2498
209.165.201.1	50.50.50.50	928	0x80000001	0x5b2f
209.165.201.1	60.60.60.60	1265	0x800003c3	0xf49b
192.0.2.0	40.40.40.40	943	0x80000001	0x53f3
192.0.2.0	50.50.50.50	935	0x80000001	0x26f8
192.0.2.0	60.60.60.60	930	0x80000001	0x7b51

Showing ASBR summary link states:

```
switch# show ip ospf lsdb asbr-summary
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====

ASBR Summary Link State Advertisements (Area 0.0.0.0)
```

LSID	ADV Router	Age	Seq#	Checksum
209.165.201.3	60.60.60.60	944	0x80000001	0x7179
192.0.2.1	50.50.50.50	935	0x80000001	0x516a

Showing external link states:

```
switch# show ip ospf lsd external
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====
```

AS External Link State Advertisements

LSID	ADV Router	Age	Seq#	Checksum
209.165.201.3	60.60.60.60	944	0x80000001	0x7179
192.0.2.1	50.50.50.50	935	0x80000001	0x516a

Showing database summary:

```
switch# show ip ospf lsd database-summary
OSPF Router with ID (10.1.1.1) (Process ID 1 VRF default)
=====
```

Area 0.0.0.0 database summary

LSA Type	Count
Router	2
Network	1
Inter-area Summary	1
ASBR Summary	0
NSSA External	0
Subtotal	4

Process 1 database summary

LSA Type	Count
Router	2
Network	1
Inter-area Summary	1
ASBR Summary	0
NSSA External	0
AS External	0
Total	4

Showing router LSAs:

```
switch# show ip ospf lsd router
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====
```

Router Link State Advertisements (Area 0.0.0.0)

LSID	ADV Router	Age	Seq#	Checksum	Link Count
1.1.1.2	1.1.1.2	15	0x80000004	0xf526	1

2.2.2.1	2.2.2.1	14	0x80000005	0x6c5e	2
2.2.2.2	2.2.2.2	104	0x80000004	0xf51a	1

Showing network LSAs:

```
switch# show ip ospf lsdB network
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====
```

Network Link State Advertisements (Area 0.0.0.0)

LSID	ADV Router	Age	Seq#	Checksum
1.1.1.2	1.1.1.2	141	0x80000001	0xc55e
2.2.2.2	2.2.2.2	230	0x80000001	0xa179

Showing network-summary link states:

```
sswitch# show ip ospf lsdB summary
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====
```

Inter-area Summary Link State Advertisements (Area 0.0.0.0)

LSID	ADV Router	Age	Seq#	Checksum
1.1.1.0	2.2.2.1	133	0x80000002	0xa089

Inter-area Summary Link State Advertisements (Area 0.0.0.1)

LSID	ADV Router	Age	Seq#	Checksum
2.2.2.0	2.2.2.1	133	0x80000002	0x7caa

Showing NSSA external link states:

```
switch(config-ospf-1)# show ip ospf lsdB nssa-external
OSPF Router with ID (2.2.2.1) (Process ID 1 VRF default)
=====
```

NSSA External Link State Advertisements (Area 0.0.0.1)

LSID	ADV Router	Age	Seq#	Checksum
8.8.8.0	1.1.1.2	162	0x80000003	0xc7b2

show ip ospf neighbors

Syntax

```
show ip ospf [<PROCESS-ID>] neighbors [<NEIGHBOR-ID>]
    [interface <INTERFACE-NAME>] [detail | summary]
    [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays information about OSPF neighbors.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display OSPF neighbor information for the particular OSPF process. Range: 1 to 63.

neighbors <NEIGHBOR-ID>

Select to display information about a particular neighbor, specified in IPv4 format (A.B.C.D).

interface <INTERFACE-NAME>

Select to display neighbor information only for the specified interface.

detail

Select to display detailed information for all the neighbors.

summary

Select to display summary information for the neighbors.

all-vrfs

Select to display neighbor information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF neighbors information for the default VRF:

```
switch# show ip ospf neighbors
OSPF Process ID 1 VRF default
=====
```

Total Number of Neighbors: 1

Neighbor ID	Priority	State	Nbr Address	Interface
2.2.2.2	1	FULL/DR	10.1.1.2	1/1/1

Showing OSPF neighbors information for VRF red:

```
switch# show ip ospf neighbors vrf red
OSPF Process ID 1 VRF red
=====
```

Total Number of Neighbors: 1

Neighbor ID	Priority	State	Nbr Address	Interface
1.1.1.1	1	FULL/BDR	10.1.1.1	1/1/1

Showing OSPF neighbors information for a specific neighbor:

```
switch# show ip ospf neighbors 2.2.2.2
Neighbor 2.2.2.2, interface address 10.1.1.2
-----
Process ID 1 VRF default, in area 0.0.0.0 via interface 1/1/1
Neighbor priority is 1, State is FULL
Options is 0x42
Dead timer due in 00:00:32
Time since last state change 00h:19m:17s
```

Showing OSPF neighbors information for a specific neighbor and interface:

```
switch# show ip ospf neighbors 2.2.2.2 interface 1/1/1
Neighbor 2.2.2.2, interface address 10.1.1.2
-----
Process ID 1 VRF default, in area 0.0.0.0 via interface 1/1/1
Neighbor priority is 1, State is FULL
DR is 10.1.1.2, BDR is 10.1.1.1
Options is 0x42
Dead timer due in 00:00:38
Retransmission queue length 0
Time since last state change 00h:22m:21s
```

Showing detail information for OSPF neighbors:

```
switch# show ip ospf neighbors detail
Neighbor 2.2.2.2, interface address 10.1.1.2
-----
Process ID 1 VRF default, in area 0.0.0.0 via interface 1/1/1
Neighbor priority is 1, State is FULL
DR is 10.1.1.2, BDR is 10.1.1.1
Options is 0x42
Dead timer due in 00:00:38
Retransmission queue length 0
Time since last state change 00h:22m:21s
```

Showing summary information for OSPF neighbors in default VRF:

```
switch# show ip ospf neighbors summary
OSPF Process ID 1 VRF default, Neighbor Summary
=====
```

Interface	Down	Attempt	Init	TwoWay	ExStart	Exchange	Loading	Full	Total
1/1/1	0	0	0	0	0	0	0	1	1
1/1/2	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	1	1

show ip ospf routes

Syntax

```
show ip ospf [<PROCESS-ID>] routes
             [<IPV4-ADDR>/<MASK>] [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```


Description

Displays OSPF routing table information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display information on the OSPF routing table for the particular OSPF process. Range: 1 to 63.

<IPV4-ADDR>

Specify an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255. .

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 32.

all-vrfs

Select to display OSPF routing table information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF routing table information:

```
switch# show ip ospf routes
Codes: i - Intra-area route, I - Inter-area route
       E1 - External type-1, E2 - External type-2

OSPF Process ID 1 VRF default, Routing Table
-----

Total Number of Routes : 2

10.1.1.0/24      (i) area: 0.0.0.0
    directly attached to interface 1/1/1, cost 1 distance 110
20.1.1.0/24      (I)
    via 10.1.1.2 interface 1/1/1, cost 2 distance 110
```

Showing OSPF routing table information for a specific subnet:

```
switch# show ip ospf routes 10.1.1.0/2
Codes: i - Intra-area route, I - Inter-area route
       E1 - External type-1, E2 - External type-2
```

```
OSPF Process ID 1 VRF default, Routing Table for prefixes 10.1.1.0/24
```

```
-----  
Total Number of Routes : 1
```

```
10.1.1.0/24      (i) area: 0.0.0.0  
    directly attached to interface 1/1/1, cost 1 distance 110
```

show ip ospf statistics

Syntax

```
show ip ospf [<PROCESS-ID>] statistics [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays OSPF statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF Process ID to display information on the OSPF statistics for the particular OSPF process.
Range: 1 to 63.

all-vrfs

Select to display OSPF statistics information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF statistics:

```
switch# show ip ospf statistics  
OSPF Process ID 1 VRF default, Statistics (cleared 1h 16m 24s ago)
```

```
-----  
Unknown Interface Drops           : 0  
Unknown Virtual Interface Drops   : 0  
Bad Instance ID Drops             : 0  
Bad IP Header Length Drops        : 0  
Wrong OSPF Version Drops          : 0  
Bad Source IP Drops               : 0  
Resource Failure Drops            : 0
```

```
Bad Header Length Drops      : 0
Total Drops                  : 0
```

show ip ospf statistics interface

Syntax

```
show ip ospf [<PROCESS-ID>] statistics interface [<INTERFACE-NAME>]
            [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays OSPF statistics for the OSPF-enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display OSPF-enabled interface statistics information on the specified OSPF process. Range: 1 to 63.

<INTERFACE-NAME>

Select to display information only for the specified interface.

all-vrfs

Select to display OSPF-enabled interface statistics information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF-enabled interfaces information:

```
switch# show ip ospf statistics interface 1/1/1
OSPF Process ID 1 VRF default, interface 1/1/1 statistics (cleared 0h 30m 28s ago)
=====
```

Tx Hello Packets	: 101	Rx Hello Packets	: 99
Tx Hello Bytes	: 101	Rx Hello Bytes	: 99
Tx DD Packets	: 101	Rx DD Packets	: 99
Tx DD Bytes	: 101	Rx DD Bytes	: 99
Tx LS Request Packets	: 101	Rx LS Requests Packets	: 99
Tx LS Request Bytes	: 101	Rx LS Request Bytes	: 99
Tx LS Update Packets	: 101	Rx LS Update Packets	: 99
Tx LS Update Bytes	: 101	Rx LS Update Bytes	: 99
Tx LS Ack Packets	: 101	Rx LS Ack Packets	: 99
Tx LS Ack Bytes	: 101	Rx LS Ack Bytes	: 99

```
Total Number of State Changes : 8
Number of LSAs                  : 29
LSA Checksum Sum                : 2345
Total Transmit Failures         : 29
Total OSPF Packets Discarded    : 999
```

Reason	Packets Dropped
Invalid type	19
Invalid length	9
Invalid checksum	0
Invalid version	23
Bad or unknown source	67
Area mismatch	1
Self-originated	19
Duplicate router ID	9
Interface standby	0
Total Hello packets dropped	60
Network Mask mismatch	10
Hello interval mismatch	10
Dead interval mismatch	10
Options mismatch	10
MTU mismatch	10
Neighbor ignored	10
Authentication errors	12
Type mismatch	6
Authentication failures	6
Wrong protocol	0
Resource failures	0
Bad LSA length	0
Others	0

```
Total LSAs Ignored : 176
Bad Type            : 10
Bad Length          : 56
Invalid Data        : 55
Invalid Checksum    : 55
```

show ip ospf virtual-links

Syntax

```
show ip ospf [<PROCESS-ID>] virtual-links [brief] [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays the current state and parameters of the OSPF virtual links.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPF process ID to display information on the OSPF virtual links for the particular OSPF process. Range: 1 to 63.

brief

Select to display brief overview information for the OSPF virtual links.

all-vrfs

Select to display OSPF virtual links information for all VRFs.

vrf <VRF-NAME>

Specify the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPF virtual links information:

```
switch# show ip ospf virtual-links
Virtual link to router 40.40.40.40 is up
-----

Process ID 21 VRF default, Transit area 0.0.0.1
Transit delay 1 sec
Timer Intervals: hello 10, dead 40, retransmit 5
No authentication
Number of Link LSAs: 0, checksum sum 0
4 state changes
```

Showing brief overview information for OSPF virtual links:

```
switch# show ip ospf virtual-links brief
OSPF Process ID 1 VRF default
=====

Total Number of Virtual Links: 1

Remote Router      Transit Area      Status
-----
2.2.2.2            0.0.0.1           down
```

summary-address

Syntax

`summary-address <IPV4-ADDR>/<MASK> [no-advertise | tag <TAG-VALUE>]`

`no summary-address <prefix/length> [no-advertise | tag <tag-value>]`

Description

Summarizes the external routes with the matching address and mask. When advertising this route, its metric is set to the lowest cost path from among the routes that were summarized.

The `no` form of this command disables route summarization.



NOTE: This command only works for an ASBR (Autonomous System Boundary Router).

Command context

config-ospf-<PROCESS-ID>

Parameters

<IPV4-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 32.

no-advertise

Do not advertise the aggregate route. Suppress routes that match the specified prefix/mask pair.

tag <TAG-VALUE>

Specify the tag for the aggregate route. The summary prefix will be advertised along with the tag value in External LSAs. Range: 0 to 4294967295

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF route summarization:

```
switch(config)# router ospf 1
switch(config-ospf-1)# summary-address 10.1.0.0/16
```

Disabling route summarization:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no summary-address 10.1.0.0/16
```

timers lsa-arrival

Syntax

timers lsa-arrival <DELAY>

no timers lsa-arrival

Description

Configures the minimum delay between receiving the same LSA from a peer. The same LSA is an LSA that contains the same LSA ID number, LSA type, and advertising router ID. If an instance of the same LSA arrives sooner before the delay expires, the LSA is dropped. Generally, the LSA arrival timer should be set to a value less than or equal to the start-time value for the command `timers throttle lsa start` on the neighbor.

The `no` form of this command sets the LSA timers to default values.

Command context

config-ospf-<PROCESS-ID>

Parameters

<DELAY>

Specifies the delay in milliseconds. Range: 0 to 600000. Default: 1000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the LSA arrival timer:

```
switch(config)# router ospf 1
switch(config-ospf-1)# timers lsa-arrival 10
```

Setting the LSA arrival timer to default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no timers lsa-arrival
```

timers throttle lsa

Syntax

```
timers throttle lsa start-time <START-TIME> hold-time <HOLD-TIME> max-wait-time <WAIT-TIME>
```

```
no timers throttle lsa
```

Description

Configures the timers for LSA generation.

The `no` form of this command sets the LSA timers to default values.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

start-time <START-TIME>

Specifies the initial wait time in milliseconds after which LSAs are generated. When set to 0, the LSAs are generated without any delay. Range: 0 to 600000. Default: 5000.

hold-time <HOLD-TIME>

Specifies the amount of time, in milliseconds, between regeneration of an LSA. The hold time doubles each time the same LSA must be regenerated, until `max-wait-time` is reached. When set to 0, LSA regeneration time is not increased. Range: 0 to 600000. Default: 0.

max-wait-time <WAIT-TIME>

Specifies the maximum wait time, in milliseconds, for regeneration of the same LSA. When set to 0, LSA regeneration time is not increased. Range: 0 to 600000. Default: 0.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the LSA timers:

```
switch(config)# router ospf 1
switch(config-ospf-1)# timers throttle lsa start-time 100 hold-time 1000 max-wait-time 10000
```

Setting LSA timers to default values:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no timers throttle lsa
```

timers throttle spf

Syntax

```
timers throttle spf start-time <START-TIME> hold-time <HOLD-TIME>
max-wait-time <WAIT-TIME>
```

```
no timers throttle spf
```

Description

Configures timers for SPF calculation. There are three timers:

- `start-time` Is the initial delay before an SPF calculation is started. Default is 200 milliseconds.
- `hold-time` Is the progressive backoff time to wait before next scheduled SPF calculation. Default is 1000 milliseconds. If a route change event occurs during this period, the value doubles until it reaches the *max-wait-time*.
- `max-wait-time` Is the maximum time to wait before the next scheduled SPF calculation. Default is 5000 milliseconds. This is used to limit the SPF hold timer and also defines the time to be considered for which the OSPF LSDB has to be stable, after which the SPF throttle mechanism is reset.

The `no` form of this command sets all the configured non-default timers to default value.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<START-TIME>

Time in milliseconds to set timer for initial SPF delay. Default: 200.

<HOLD-TIME>

Time in milliseconds to set the minimum hold time between two consecutive SPF calculations. Default: 1000.

<WAIT-TIME>

Time in milliseconds to set the maximum wait time between two consecutive SPF calculations. Default: 5000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting non-default timer values for SPF throttling:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# timers throttling spf start-time 500 hold-time 3000 max-wait-time 9000
Switch(config-ospfv3-1)# show running-config current-context
router ospfv3 1
```



```
area 0.0.0.0
area 0.0.0.1
area 0.0.0.2 nssa no-summary
area 0.0.0.3 stub
```

Setting default timer values for SPF throttling after configuring non-default values:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no timers throttling spf
```

transit-delay

Syntax

```
transit-delay <SECONDS>
```

```
no transit-delay
```

Description

Sets the time delay in Link state transmission for virtual links.

The `no` form of this command sets the delay in Link state transmission to the default of 1 second for virtual links.

Command context

```
config-router-vlink
```

Parameters

<SECONDS>

Specifies the time delay for the transit delay, in seconds. Default: 1 second. Range: 1-3600.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv2 virtual links transit delay:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# transit-delay 30
```

Setting OSPFv2 virtual links transit delay to default:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink)# no transit-delay
```

trap-enable

Syntax

```
trap-enable
```

```
no trap-enable
```

Description

Enables the notification of the events to be sent as traps to the SNMP management stations for OSPF.

The `no` form of this command disables the notification of the events to be sent as traps to the SNMP management stations for OSPF.

Command context

```
config-ospf-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling sending notification of events as traps:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# trap-enable
```

Disabling sending notification of events as traps:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no trap-enable
```

Overview

OSPFv3 is the IPv6 implementation of Open Shortest Path First protocol (OSPFv2 is the IPv4 implementation of this protocol). OSPFv3 is a routing protocol which is described in RFC5340 entitled OSPF for IPv6. It is a link-state based IGP (Interior Gateway Protocol) routing protocol. It is widely used with medium to large-sized enterprise networks.

The characteristics of OSPFv3 are:

- Provides a loop-free topology using SPF algorithm.
- Allows hierarchical routing using area 0 (backbone area) as the top of the hierarchy.
- Supports load balancing with equal cost routes for same destination.
- OSPFv3 is a classless protocol and allows for a hierarchical design with VLSM (Variable Length Subnet Masking) and route summarization.
- Scales enterprise size network easily with area concept.
- Provides fast convergence with triggered, incremental updates through Link State Advertisements (LSAs).

Some OSPFv3 configuration is done in the configuration context, others in the OSPFv3 router context, or in the interface context. OSPFv3 can be configured on routed ports, VLAN interfaces, LAG interfaces, and loopback interfaces. All such configurations work in the mentioned interfaces context. OSPFv3 mandates the associated interface to be routing interface.

The switch supports congestion control (prioritizing hello packets, inactivity timer reset, and adjacency throttling).

Supported features

- OSPFv3 neighbor adjacency, Hello protocol, multiple areas, Inter-area routing
- AS external routes, stub areas, totally stubby areas, NSSA, ABR, ASBR
- Designated Router/Backup Designated Router
- Point-to-point interfaces/broadcast interfaces
- Virtual Links
- Bidirectional Forwarding Detection (BFD) - refer to the *High Availability Guide* for additional information
- Equal-cost multipath
- Null authentication, Simple password authentication, MD5 authentication
- Area range aggregation - Type-3/Type-7
- External route aggregation - Type-5 LSAs
- Graceful Restart - un-planned, restart interval, Graceful Restart Helper
- Stub router advertisement

- SPF throttling
- LSA Throttling
- SPF throttling
- Graceful Restart - un-planned, restart interval, Graceful Restart Helper
- Stub router advertisement
- Configuration of interface parameters such as priority, cost, hello-interval, dead-interval, retransmit-interval, transit-delay, etc.
- Configuration of virtual link parameters such as hello-interval, dead-interval, retransmit-interval, transit-delay, etc.
- Route redistribution
- Passive interfaces
- Multi VRF support with each VRF having up to eight OSPF process instances
- Congestion control (prioritizing hello packets, inactivity timer reset, and adjacency throttling)

How OSPFv3 protocol works

OSPFv3 protocol

The protocol uses Link State Advertisements (LSAs) transmitted by each router to update neighboring routers regarding its interfaces and the routes available through those interfaces. Each routing switch in an area also maintains a link-state database (LSDB) that describes the area topology. (All routers in a given OSPF area have identical LSDBs.) The routing switches used to connect areas to each other flood summary link LSAs and external link LSAs to neighboring OSPF areas to update them regarding available routes. Through this means, each OSPF router determines the shortest path between itself and a desired destination router in the same OSPF domain (Autonomous System (AS)).

OSPFv3 concepts

OSPFv3 is a link-state routing protocol applied to IPv6 routers grouped into OSPFv3 areas identified by the IPv6 routing configuration on each routing switch. Each OSPFv3 area includes one or more networks. OSPFv3 routers use hello packets and LSAs to maintain OSPFv3 operation across networks within an area and between areas within an OSPFv3 domain.

OSPFv3 uses hello packets to initiate and preserve relationships between neighboring routers on the same interface.

OSPFv3 Link-state advertisement (LSA) types

OSPFv3 uses LSAs transmitted by each router to update neighboring routers regarding its interfaces and the routes available through those interfaces.

Each routing switch in an area also maintains an LSDB that describes the area topology. (All routers in a given OSPFv3 area have identical LSDBs, and each router uses the LSDB to build its own shortest-path tree.)

The routing switches used to connect areas to each other flood inter-area-prefix-LSAs, inter-area-router-LSAs, and AS-external-LSAs to backbone area to update it regarding available routes. Each OSPFv3 router determines the shortest path between itself and a desired destination router in the same OSPFv3 domain (AS). Routed traffic in an OSPFv3 AS is classified as one of the following:

- Intra-area traffic
- Inter-area traffic
- External traffic

The routing switches support the LSAs listed in the following table.

Link-state type	Description	Use	Flood scope
0x2001	Router-LSA	Describes the state of each active interface on a router for Area a given area. (Excludes loopback interfaces and interfaces that have not achieved full adjacency.)	Area
0x2002	Network-LSA	Describes the OSPFv3 routers in a given network.	Area
0x2003	Inter-area-prefix-LSA	Describes the route to a prefix in another OSPFv3 area of Area the same AS. (Excludes prefixes for link-local addresses.) Propagated through backbone area to other areas.	Area
0x2004	Inter-area-router-LSA	Describes the route to an ASBR in another OSPFv3 normal area (including the backbone area) of the same AS. Propagated through backbone area to other areas. (Excludes any ASBR in the same area as the router sending the LSA.)	Area
0x4005	AS-external-LSA	Describes the route to a destination prefix in another AS (external route). (Excludes prefixes for link-local addresses.) Originated by ASBR in normal or backbone areas of an AS and propagates through backbone area to other normal areas. Does not flood over virtual links and is not summarized in virtual links. For injection into an NSSA, an NSSA ABR generates a type-7-default-LSA advertising the default route (::/0).	AS
0x2007	NSSA-LSA	Describes the route to a destination in another AS (external route). Originated by ASBR in NSSA. ABR translates type-7 LSAs to AS-external-LSAs for injection into the backbone area.	NSSA

Table Continued

Link-state type	Description	Use	Flood scope
0x0008	Link-LSA	For other routers on the same VLAN interface (except virtual links), describes the router link-local address and any other IPv6 prefixes reachable on the VLAN. Link LSAs are not flooded over virtual links.	Link-local
0x2009	Intra-area-prefix-LSA	Generated on transit links within an area by the DR operating on those links. Also, every OSPFv3 router generates this LSA to refer to stub and loopback prefixes on the router.	Area

OSPFv3 area types

OSPFv3 is built upon a hierarchy of network areas. All areas for a given OSPFv3 domain reside in the same AS. An AS is defined as a number of contiguous networks that share an interior gateway routing protocol.

An AS can be divided into multiple areas, including the backbone (area 0). Because each area represents a collection of contiguous networks and hosts, the topology of a given area is not known by the internal routers in any other area. Areas define the boundaries to which router-LSAs and network-LSAs are broadcast. This functionality limits the amount of LSA flooding that occurs within the AS. It also helps to control the size of the link-state databases (LSDBs) maintained in OSPFv3 routers.

An area is represented in OSPFv3 by either a 32-bit dotted-decimal address or a number. Area types include: Backbone, Normal, Not-so-stubby (NSSA), and Stub.

Backbone area

Every AS must have one (and only one) backbone area (identified as area 0 or 0.0.0.0.) The ABRs of all other areas in the same AS connect to the backbone area, either physically through an ABR or through a configured, virtual link. The backbone is a special type of area and serves as a transit area for carrying the inter-area-prefix-LSAs, AS-external-LSAs, and routed traffic between non-backbone areas, as well as the router-LSAs, network-LSAs, and routed traffic internal to the area. ASBRs are allowed in backbone areas.

Normal area

A normal area allows inter-area-prefix-LSAs and AS-external-LSAs to and from the backbone area. A normal area connects to the AS backbone area through one or more ABRs (physically or through a virtual link) and allows router-LSAs and network LSAs within this area. ASBRs are allowed in normal areas.

Stub area

Stub area connects to the AS backbone through one or more ABRs. It does not allow an internal ASBR, and does not allow AS-external-LSAs. A stub area supports these actions:

- Advertise the area summary routes to the backbone area.
- Advertise inter-area routes from other areas.
- Use the inter-area-prefix-LSA default route to advertise routes to an ASBR and to other areas.

You can configure the stub area ABR to do the following:

- Suppress advertising on the area's summarized internal routes into the backbone area.
- Suppress LSA traffic from other areas in the AS by replacing inter-area-prefix-LSAs and the default external route from the backbone area with the default summary route (::/0.). This area is called totally stubby area.

Virtual links are not allowed for stub areas.

Not-so-stubby (NSSA) area

An NSSA area connects to the backbone area through one or more ABRs. NSSAs are used where an ASBR exists in an area where you want to:

- Block injection of external routes from other areas of the AS.
- Advertise type-7-LSA external routes (learned from the ASBR) to the backbone area as AS-external-LSAs.

NSSAs also support the following:

- Advertise inter-area-prefix-LSAs from the backbone area into the NSSA. (If no-summary is enabled, the NSSA ABR suppresses these LSAs from the backbone and, instead, injects the default route into the NSSA.)
- Advertise NSSA inter-area-prefix-LSAs to the backbone area.

Virtual links are not allowed for NSSAs.

OSPFv3 router types

Internal routers

Internal OSPFv3 routers belong to only one area. Internal routers flood router-LSAs to all routers in the same area and maintain identical LSDBs.

Area border routers (ABRs)

Area border routers have membership in multiple areas. ABRs are used to connect the various areas in an AS to the backbone area for that AS. Multiple ABRs can be used to connect a given area to the backbone, and a given ABR can belong to multiple areas other than the backbone.

An ABR maintains a separate LSDB for each area to which it belongs. (All routers within the same area have identical LSDBs.) The ABR is responsible for flooding inter-area-prefix-LSAs and inter-area router LSAs between its border areas.

You can reduce summary LSA flooding by configuring area ranges. An area range enables you to assign an aggregate address to a range of IPv6 addresses. This aggregate address is advertised instead of all the individual addresses it represents. You can assign up to eight ranges in an OSPFv3 area.

Autonomous system boundary router (ASBR)

Autonomous system boundary routers run one or more interior gateway protocols and serve as a gateway to other autonomous systems operating with interior gateway protocols. The ASBR imports and translates different protocol routes into OSPFv3 through redistribution. ASBRs can be used in backbone areas, normal areas, and NSSAs, but not in stub areas.

Designated routers (DRs)

In an OSPFv3 network having two or more routers, one router is elected to serve as the designated router (DR) and another router to act as the backup designated router (BDR). All other routers in the same network segment forward their routing information to the DR and BDR, and the DR forwards this information to all

routers in the network. This functionality minimizes the amount of repetitive information that is forwarded on the network by eliminating the need for each individual router in the area to forward its routing information to all other routers in the network. If the area includes multiple networks, each network elects its own DR and BDR.

In an OSPFv3 network with no DR and no BDR, the neighboring router with the highest priority is elected the DR and the router with the next highest priority is elected the BDR. If the DR goes off-line, the BDR automatically becomes the DR, and the router with the next highest priority then becomes the new BDR. If multiple routing switches on the same OSPFv3 network are declaring themselves DRs, both priority and router ID are used to select the DR and BDR.

Priority is configurable using the `ipv6 ospfv3 priority` command at the interface level. You can use this command to help bias one router as the DR. If two neighbors share the same priority, the router with the highest router ID is designated the DR. The router with the next highest router ID is designated the BDR. Routers retain DR/BDR states throughout the period for which the adjacency is up and running.

OSPFv3 configuration task list

Tasks at a glance

- [Configuring OSPFv3 on the routing switch](#)
- [Assigning the routing switch to an OSPF area](#)
- [Setting OSPFv3 network for the area](#)
- [Configuring external route redistribution and control](#)
- [Configuring area ranges on an ABR to reduce advertisements to the backbone](#)
- [Influencing route choice by changing the administrative distance](#)
- [Configuring graceful restart](#)
- [Configuring OSPFv3 virtual link settings](#)
- [Configuring OSPFv3 interface settings](#)
- [Configuring BFD for OSPFv3](#)
- [Configuring all OSPFv3 interfaces as passive](#)
- [Viewing OSPFv3 information](#)
- [Clearing OSPFv3 statistics on a switch](#)

Configuring OSPFv3 on the routing switch

Create the OSPFv3 instance and enter the OSPFv3 configuration context. From this context, you can proceed with other OSPFv3 configuration.

Prerequisites

- You must be in the global configuration context, as indicated by the `switch(config) #` prompt to create the OSPFv3 instance and enter the OSPFv3 configuration context.
- To configure a router ID, and other OSPFv3 configuration you must be in the OSPFv3 router configuration context, as indicated by the `switch(config-ospfv3-1) #` prompt.

Procedure

1. Create the OSPFv3 instance and enter the OSPFv3 configuration context using the following command. For command details, see [router ospfv3](#).

```
router ospfv3 <PROCESS-ID> [vrf <VRF-NAME>]
```

For example, the following command creates OSPF instance 1.

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)#
```

2. Configure a global router ID using the following command. For command details, see [router-id](#).

```
router-id <ROUTER-ADDRESS>
```

For example, the following command sets the router ID to 1.1.1.1.

```
switch(config-ospfv3-1)# router-id 1.1.1.1
```

3. Optionally, if the OSPFv3 process was disabled (it is enabled by default), enable the OSPFv3 process using the following command.

```
enable
```

For command details, see [enable](#). (Refer to [disable](#) for disabling the OSPFv3 process).

Creating an OSPFv3 area

Create a Normal, Stub, or Not So Stubby (NSSA) area.

Prerequisites

You must be in the OSPFv3 router configuration context.

Procedure

1. Create an OSPFv3 area for the routing switch using one of the following commands:

- Create a Normal area using the following command: `area <area-id>`. For command details, see .
- Create a Stub area using the following command: `area <area-id> stub`. For command details, see .
- Create a Not So Stubby Area using the command: `area <area-id> nssa`.

For example, the following command creates a normal area with an area identifier of 10.1.1.1. Area identifier could alternatively be entered in decimal format such as 1.

```
switch(config-ospfv3-1)# area 1.1.1.1
```

Setting OSPFv3 network for the area

After you define an OSPFv3 area, you can assign one or more networks to it. The OSPFv3 protocol will run on the interface with the configured link-local ipv6 address. The interfaces that have an IPv6 address configured in this network or in a subset of this network will participate in the OSPFv3 protocol.

Prerequisites

Routing must be enabled on the interface where you are planning to enable OSPFv3.

You must be in the interface configuration context, as indicated by the `switch(config-if) #` prompt.

Procedure

1. Set an OSPFv3 network for the area using the following command. For command details, see [ipv6 ospfv3 area](#).

```
ipv6 ospfv3 <PROCESS-ID> area <AREA-ID>
```

For example, use the following command to assign interface 1/1/1 to OSPFv3 area 1. The area can alternatively be entered in IO address format.

```
switch(config) # interface 1/1/1
switch(config-if) # ipv6 ospfv3 1 area 1
```

2. Optionally, you can disable OSPFv3 on the interface using the following command. For command details, see [ipv6 ospfv3 shutdown](#).

```
ipv6 ospfv3 shutdown
```

```
switch(config) # interface 1/1/1
switch(config-if) # ipv6 ospfv3 shutdown
```

Configuring external route redistribution and control

Configuring route redistribution for OSPFv3 establishes the routing switch as an ASBR for importing and translating different protocol routes from other IGP domains into an OSPFv3 domain. When you configure redistribution for OSPFv3, you can specify that static, or connected routes external to the OSPFv3 domain are imported as OSPFv3 routes.

Prerequisites

You must be in the OSPFv3 router configuration context, as indicated by the `switch(config-ospfv3-1) #` prompt.

Procedure

1. Enable route redistribution using the following command and specify one of the following parameters: connected and static. For command details, see [redistribute](#).

```
redistribute {connected | static | ripng | ospfv3 <PROCESS-ID>} [route-map <ROUTE-MAP-NAME>]
```

For example, the following command sets redistribution of connected routes as OSPF routes:

```
switch(config-ospfv3-1) # redistribute connected
```

2. Optionally, modify the default metric for redistribution using the following command. For command details, see [default-metric](#).

```
default-metric <METRIC-VALUE>
```

For example, the following command sets the default metric for redistribution to 37.

```
switch(config-ospfv3-1) # default-metric 37
```

3. Optionally, set the cost of default-summary LSAs using the following command. For command details, see .

```
area <AREA-ID> default-metric <METRIC>
```

For example, the following command sets the cost of default summary LSAs to 2.

```
switch(config-ospfv3-1) # area 1 default-metric 2
```

4. Optionally, set the protocol to advertise a maximum metric so that other routers do not prefer this router as an intermediate hop in their shortest path first (SPF) calculations. For command details, see [max-metric router-lsa](#).

```
max-metric router-lsa [on-startup [<INTERVAL>]]
```

For example, the following command sets advertise max-metric router-lsa on startup.

```
switch(config-ospfv3-1) # max-metric router-lsa on-startup 3000
```

5. Set the maximum number of ECMP routes that OSPFv3 can support using the following command. For command details, see [maximum-paths](#).

```
maximum-paths <MAXIMUM>
```

For example, the following command sets the maximum number of ECMP routes to 8.

```
switch(config-ospfv3-1) # maximum-paths 8
```

Configuring area ranges on an ABR to reduce advertisements to the backbone

You can configure area ranges to reduce inter-area advertisements by summarizing a range of IP addresses into a single route advertisement. This action prevents an ABR from advertising specific networks or subnets to the backbone area.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-ospfv3) #` prompt.

Procedure

Summarize inter-area or NSSA paths using the following command. For command details, see .

```
area <area-id> range <ip-prefix> type {inter-area | nssa} [no-advertise]
```

For example, use the following command to summarize routes matching the area range 172.77.114.0/24 using inter-area as the type of address aggregation.

```
switch(config) # router ospfv3 1  
switch(config-ospfv3-1) # area 1 range fd00::/64 type inter-area
```

In another example, use the following command to specify DoNotAdvertise status for routes matching the area range fd00::/64. Use nssa as the type of address aggregation.

```
switch(config) # router ospfv3 1  
switch(config-ospfv3-1) # area 1 range fd00::/64 type inter-area no-advertise
```

Influencing route choice by changing the administrative distance

The administrative distance is used to influence the selection of routes learned by different protocols.

Prerequisites

You must be in the OSPFv3 router configuration context, as indicated by the `switch(config-ospfv3-1) #` prompt.

Procedure

Reconfigure the administrative distance using the following command. For command details, see [distance](#).

```
distance <distance>
```

For example, use the following command to set administrative distance to 100.

```
switch(config-ospfv3-1) # distance 100
```

Configuring graceful restart

OSPFv3 routing can be gracefully restarted on switches without losing packets that are in transit. There is no effect on the saved switch configuration.

Prerequisites



NOTE: Graceful restart is only applicable to the 6300-VSF and 6400, not for 832x switches.

You must be in the OSPFv3 router configuration context, as indicated by the `switch(config-ospfv3-1) #` prompt.

Procedure

Configure graceful restart of using the following command. For command details, see [graceful-restart](#).

```
graceful-restart [restart-interval <seconds> | helper]
```

For example, the following command specifies 50 seconds as the maximum interval another router will wait for this router to gracefully restart.

```
switch(config-ospfv3-1) # graceful-restart restart-interval 50
```

Configuring OSPFv3 virtual link settings

The OSPF interface parameters for this process are automatically set to their default values for virtual links. You can optionally adjust the following OSPF virtual link settings.

Prerequisites

You must be in the router vlink configuration context, as indicated by the `switch(config-router-vlink) #` prompt.

Procedure

- Set the time interval between OSPF hello packets for the OSPF virtual links using the following command. For command details, see [hello-interval](#).

```
hello-interval <seconds>
```

For example:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1  
switch(config-router-vlink)# hello-interval 30
```

- Set the interval after which a neighbor is declared dead if no hello packet is received on the OSPF virtual links. Use the following command. For command details, see [dead-interval](#)

```
dead-interval <seconds>
```

- Set the time between retransmitting lost link state advertisements for the OSPF virtual links using the following command. For command details, see [retransmit-interval](#).

```
retransmit-interval <seconds>
```

- Sets the transit delay in Link state transmission for the OSPF virtual links using the following command. For command details, see [transit-delay](#).

```
transit-delay <seconds>
```

Configuring OSPFv3 interface settings

You can optionally adjust the following OSPFv3 interface settings.

Prerequisites

You must be in the interface configuration context, as indicated by the `switch(config-if)#` prompt.

Procedure

- Set the interface cost using the following command. For command details, see [ipv6 ospfv3 cost](#).

```
ipv6 ospfv3 cost <interface-cost>
```

For example, the following command sets the cost associated with interface 1/1/1 to 100.

```
switch(config)# interface 1/1/1  
switch(config-if)# ipv6 ospfv3 cost 100
```

- Set the time interval between OSPFv3 hello packets for the OSPFv3 interface using the following command. For command details, see [ipv6 ospfv3 hello-interval](#).

```
ipv6 ospfv3 hello-interval <seconds>
```

- Set the interval after which a neighbor is declared dead if no hello packet is received on the OSPFv3 interface using the following command. For command details, see [ipv6 ospfv3 dead-interval](#).

```
ipv6 ospfv3 dead-interval <seconds>
```

- Set the time between re-transmitting lost link state advertisements for the OSPFv3 interface using the following command. For command details, see [ipv6 ospfv3 retransmit-interval](#).

```
ipv6 ospfv3 retransmit-interval <seconds>
```

- Sets the time delay in Link state transmission for the OSPFv3 interface using the following command. For command details, see [ipv6 ospfv3 transit-delay](#).

```
ipv6 ospfv3 transit-delay <seconds>
```

- Set the OSPFv3 network type for the interface. For command details, see [ipv6 ospfv3 network](#).

```
ipv6 ospfv3 network {broadcast|point-to-point}
```

- Set the OSPFv3 priority on the interface using the following command. For command details, see [ipv6 ospfv3 priority](#).

```
ipv6 ospfv3 priority <number-value>
```

- Set the OSPFv3 interface as OSPFv3 passive interface. With this setting the interface participates in OSPFv3 but does not send or receive packets on that interface. For command details, see [ipv6 ospfv3 passive](#).

```
ipv6 ospfv3 passive
```

Configuring BFD for OSPFv3

Enabling BFD Support for OSPFv3 over IPv6 enables OSPFv3 to register with Bidirectional Forwarding Detection (BFD) to receive forwarding path detection failure messages. You can either configure BFD for OSPFv3 globally on all interfaces or configure it selectively on one or more interfaces. BFD creates a session in asynchronous mode as soon as the switch reaches the 2-Way state with a neighbor. Once the session is established BFD will commence sending echos for path failure detection.

There are two methods for enabling BFD for OSPFv3:

1. Enable BFD for all interfaces enabled for OSPFv3 by using the BFD all-interfaces command in router configuration mode
2. Enable BFD for a subset of interfaces that have OSPFv3 enabled by using the IPv6 ospfv3 BFD command in interface configuration mode



NOTE: OSPFv3 needs to be enabled prior to enabling BFD on one or more interfaces.

Examples

Enable BFD on all OSPFv3 Interfaces

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# bfd all-interfaces
```

Disable BFD on all OSPFv3 Interfaces

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no bfd all-interfaces
```

Enable BFD on a single OSPFv3 interface

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ospfv3 bfd
```

Disable BFD on a single OSPFv3 interface

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ospfv3 bfd disable
```

Set BFD as default on an OSPFv3 interface

```
switch(config)# interface 1/1/1
switch(config-if)# no ipv6 ospfv3 bfd
```

Configuring all OSPFv3 interfaces as passive

OSPFv3 sends LSAs to all other routers in the same AS. To limit the flooding of LSAs throughout the AS, you can configure OSPFv3 to be passive.

Prerequisites

You must be in the OSPFv3 router configuration context, as indicated by the `switch(config-ospfv3-1) #` prompt.

Procedure

Configure all OSPFv3 interfaces as passive using the following command. For command details, see [passive-interface default](#).

```
passive-interface default
```

```
switch(config-ospfv3-1) # passive-interface default
```

Configuring SPF throttling timers

SPF calculation is throttled with default timer values (start-time 200ms, hold-time 1000ms, max-wait-time 5000ms). You can throttle SPF calculation by configuring non-default timers to improve performance of a specific network configuration.

Prerequisites

You must be in the router configuration context, as indicated by the `switch(config-router) #` prompt.

Procedure

Reconfigure SPF throttling timers using the following command. For command details, see [timers throttle spf](#)

```
timers throttle spf start-time <milliseconds> hold-time <milliseconds> max-wait-time <milliseconds>
```

For example, use the following command to set start-time to 500ms, hold-time to 3000ms and max-wait-time to 9000ms:

```
switch(config-ospf-1) # timers throttle spf start-time 500 hold-time 3000 max-wait-time 9000
```

Viewing OSPFv3 information

Prerequisites

Use these show commands from the Manager context, as indicated by the `switch#` prompt.

Procedure

1. To view OSPFv3 information, use the following commands. For command details and examples, click the links.
 - To view general OSPFv3, area, state and configuration information use: [show ipv6 ospfv3](#).

- To view information about OSPFv3 interfaces use: **show ipv6 ospfv3 interface.**
- To view information about OSPFv3 neighbors use: **show ipv6 ospfv3 neighbors.**
- To view OSPFv3 routing table information use: **show ipv6 ospfv3 routes.**
- To view OSPFv3 statistics for the OSPFv3 interfaces use **show ipv6 ospfv3 statistics interface.**

Clearing OSPFv3 statistics on a switch

Clear OSPFv3 event statics using the following command. For command details, see **clear ipv6 ospfv3 statistics**

```
clear ipv6 ospfv3 [<PROCESS-ID>] statistics [interface [<INTERFACE-NAME>]] [all-  
vrfs | vrf <VRF-NAME>]
```

For example, the following command clears OSPFv3 event statistics from interface 1/1/1.

```
switch# clear ipv6 ospfv3 statistics interface 1/1/1
```

OSPFv3 commands

area

Syntax

```
area <AREA-ID>
```

```
no area <AREA-ID>
```

Description

Creates a normal area with <AREA-ID> set if not present. If area is present and is not the normal area, this command changes the area type to normal area.

The **no** form of this command deletes the area with the <AREA-ID> specified. The area can be of any type (stub, stub no-summary, and default normal area).

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- Area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- Area identifier in decimal format. Range: 0 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating a normal area:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1
switch(config-ospfv3-1)# area 1.1.1.1
```

Deleting an area:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no area 1
switch(config-ospfv3-1)# no area 1.1.1.1
```

area authentication ipsec

Syntax

```
area <AREA-ID> authentication ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>
```

```
no area <AREA-ID> authentication
```

Description

Configures IPsec AH authentication for the specified area. OSPFv3 interfaces which have IPsec configured at the interface context will not use area level IPsec.

The `no` form of this command removes IPsec AH authentication for the specified area.



NOTE: IPsec is not supported for 6in6 tunnel interfaces.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec AH header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. The SPI must be unique on the switch. Range: 256-4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: `md5` or `sha1`.

<KEY-TYPE>

Specifies the key type to use: `plaintext` (unencrypted), `hex-string` (encrypted) or `ciphertext` (encrypted).

<AUTH-KEY>

Specifies the key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting area 1 to use IPsec authentication(AH):

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 1 authentication ipsec spi 256 sha1 plaintext abcd
```

Removing IPsec authentication (AH) on area 1:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no area 1 authentication
```

area encryption ipsec

Syntax

```
area <AREA-ID> encryption ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>  
( {<ENCR-TYPE> <KEY-TYPE> <ENCR-KEY> } | NULL)
```

```
no area <AREA-ID> encryption
```

Description

Configures IPsec ESP with authentication/encryption algorithm type and key for the specified area. OSPFv3 interfaces with IPsec configured at interface context will not use area level IPsec ESP configuration.

The **no** form of this command removes IPsec ESP from the specified area.



NOTE: IPsec is not supported for 6in6 tunnel interfaces.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (**x.x.x.x**), where **x** is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec AH header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. The SPI must be unique on the switch. Range: 256 to 4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: **md5** or **sha1**.

<ENCR-TYPE>

Specifies the algorithm to use for encryption: **3des**, **aes**, **des** or **null**.



NOTE: `aes` will be considered AES128, AES192 or AES256 based on key length.

<KEY-TYPE>

Specifies the key type to use: `plaintext` (unencrypted) `hex-string` (encrypted) or `ciphertext` (encrypted).

<AUTH-KEY>

Specifies authentication key.

<ENCR-KEY>

Specifies encryption key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting area 0 to use IPsec ESP:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 0 encryption ipsec spi 256 md5 plaintext abcd des plaintext abcdefab
```

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 0 encryption ipsec spi 256 md5 plaintext abcd null
```

Removing IPsec ESP on area 0:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no area 0 encryption
```

area nssa

Syntax

```
area <AREA-ID> nssa [no-summary]
```

```
no area <AREA-ID> nssa [no-summary]
```

Description

Creates the NSSA area (Not So Stubby Area) with `<AREA-ID>` if not present. If area is present and not NSSA area, this command changes the area type to NSSA area. If `no-summary` is used, area type will be NSSA No-Summary.

The `no` form of this command clears the NSSA area type. That is, the configured area will be changed to default normal area. The `no area <AREA-ID> nssa no-summary` command enables sending inter-area routes into NSSA, but will not unset the area as NSSA.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

nssa [no-summary]

Specifies Not So Stubby Area (NSSA) area type. If area is present and not NSSA area, parameter changes the area type to NSSA area. If `no-summary` is specified, area type will be NSSA No-Summary, which means do not inject inter-area routes into NSSA.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating an NSSA area for OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1 nssa
switch(config-ospfv3-1)# area 1 nssa no-summary
```

Clearing the NSSA area for OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no area 1 nssa
switch(config-ospfv3-1)# no area 1 nssa no-summary
```

area range

Syntax

```
area <AREA-ID> range <IP-PREFIX> type {inter-area | nssa} [no-advertise]
```

```
no area <AREA-ID> range <IP-PREFIX> type {inter-area | nssa} [no-advertise]
```

Description

Summarizes the routes with the matching address or masks for OSPFv3. This command only works for border routers.

The `no` form of this command unsets the route summarization for the configured IPv4 prefix address on the ABR. When using the `no` form of the command with the `no-advertise` option, enables advertising this range to other areas.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

range <IP-PREFIX>

Specifies summarizing routes matching the area range prefix/mask.

type {inter-area | nssa}

Specifies the type this address aggregation applies to as either inter-area range prefix or NSSA range prefix.

no-advertise

Specifies the address range status as DoNotAdvertise (do not advertise this range to other areas).

Authority

Administrators or local user group members with execution rights for this command.

Examples

Summarizing inter-area or NSSA paths on OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1 range fd00::/64 type inter-area
switch(config-ospfv3-1)# area 1 range fd00::/64 type nssa
switch(config-ospfv3-1)# area 1 range fd00::/64 type inter-area no-advertise
```

Unsetting summarization on OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no area 1 range fd00::/64 type inter-area
switch(config-ospfv3-1)# no area 1 range fd00::/64 type nssa
switch(config-ospfv3-1)# no area 1 range fd00::/64 type inter-area no-advertise
```

area stub

Syntax

area <AREA-ID> stub [no-summary]

no area <AREA-ID> stub [no-summary]

Description

Creates the stub area with <AREA-ID> if not present. If area is present and is not the stub area, this command changes the area type to stub area. If no-summary is used, area type will be totally stubby area.

The no form of this command unsets the area type as stub. The configured area will be changed to the default normal area. The no area <AREA-ID> stub no-summary command will start sending Area Border Router (ABR) summary link advertisements into the stub area, but will not unset the stub area.

Command context

config-ospfv3-<PROCESS-ID>

Parameters

<AREA-ID>

Specifies the area ID in one of the following formats.

- Area identifier in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- Area identifier in decimal format. Range: 0 to 4294967295.

stub [no-summary]

Specifies stub area type. If area is present and not stub area, this parameter changes the area type to stub area. If `no-summary` is specified, area type will be totally stubby area, which means do not inject interarea routes into stub.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating a stub area:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1 stub
switch(config-ospfv3-1)# area 1 stub no-summary
```

Unsetting the stub area type:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1) # no area 1 stub
switch(config-ospfv3-1) # no area 1 sub no-summary
```

area virtual-link

Syntax

```
area <AREA-ID> virtual-link <ROUTER-ID>
```

```
no area <AREA-ID> virtual-link <ROUTER-ID>
```

Description

Creates an OSPF virtual link with remote ABR (if not created already) and enters the vlink context.

The `no` form of this command deletes an OSPF virtual link with the specified router ID of the remote ABR. If `no <ROUTER-ID>` is specified, the `no` form of the command sets the virtual link to the default settings.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- OSPF area identifier in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.
- OSPF area identifier in decimal format. Range: 0 to 4294967295.

virtual-link <ROUTER-ID>

Configures a virtual link with the specified router ID of the remote ABR.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring OSPF virtual links:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
```

Deleting OSPF virtual links:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no area 100 virtual-link 100.0.1.1
```

area default-metric

Syntax

```
area <AREA-ID> default-metric <METRIC>
```

```
no area <AREA-ID> default-metric
```

Description

Sets the cost of default-summary LSAs announced to the stub/nssa areas.

The `no` form of this command resets the cost of the default-summary LSAs announced to stub/nssa areas to the default of 1.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<AREA-ID>

Specifies the area ID is one of the following formats.

- Area ID in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- Area ID as a decimal value. Range: 0 to 4294967295

default-metric <METRIC>

Specifies the default metric of default-summary LSAs announced to the stub/nssa areas, to the specified value. Default: 1. Range: 0 to 16777215.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting cost for default LSA summary:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 1 default-metric 2  
switch(config-ospfv3-1)# area 1.1.1.1 default-metric 2
```

Setting cost for default LSA summary to default:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no area 1 default-metric  
switch(config-ospfv3-1)# no area 0.0.0.1 default-metric
```

authentication ipsec spi

Syntax

```
authentication ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>
```

```
no authentication
```

Description

Configures IPsec AH authentication for Vlink.

The `no` form of this command removes IPsec AH authentication for the specified area.

Command context

```
config-router-vlink
```

Parameters

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec AH header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. The SPI must be unique on the switch. Range: 256 to 4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: `md5` or `sha1`.

<KEY-TYPE>

Specifies the key type to use: `plaintext` (unencrypted), `hex-string` (encrypted) or `ciphertext` (encrypted).

<AUTH-KEY>

Specifies the key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting area 1 to use IPsec authentication (AH) for Vlink:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1
switch(config-ospfv3-1)# area 1 virtual-link 3.3.3.3
switch (config-router-vlink6)# authentication ipsec spi 256 sha1 plaintext abcd
```

Removing IPsec authentication (AH) for Vlink on area 1:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1 virtual-link 3.3.3.3
switch(config-router-vlink6)# no authentication
```

clear ipv6 ospfv3 neighbors

Syntax

```
clear ipv6 ospfv3 [<PROCESS-ID>] neighbor [<NEIGHBOR>] [interface [<INTERFACE-NAME>]]
[all-vrfs | vrf <VRF-NAME>]
```


Description

Resets the neighbor and clears the OSPF neighbor information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies the OSPFv3 process ID to clear the statistics for the particular OSPFv3 process. Range: 1 to 63.

<NEIGHBOR>

Specifies the router ID of a neighbor.

<INTERFACE-NAME>

Specifies the OSPFv3 statistics to clear for the specified interface.

all-vrfs

Select to clear the OSPFv3 statistics for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Clearing the OSPFv3 neighbor information:

```
switch# clear ipv6 ospfv3 1 neighbor
switch# clear ipv6 ospfv3 1 neighbor 3.3.3.3
switch# clear ipv6 ospfv3 1 neighbor interface 1/1/1
switch# clear ipv6 ospfv3 neighbor 5.5.5.5 vrf red
```

clear ipv6 ospfv3 statistics

Syntax

```
clear ipv6 ospfv3 [<PROCESS-ID>] statistics [interface [<INTERFACE-NAME>]]
[all-vrfs | vrf <VRF-NAME>]
```

Description

Clears the OSPFv3 event statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies the OSPFv3 process ID to clear the statistics for the particular OSPFv3 process. Range: 1 to 63.

<INTERFACE-NAME>

Specifies the OSPFv3 statistics to clear for the specified interface.

all-vrfs

Select to clear the OSPFv3 statistics for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Clearing the OSPFv3 event statistics:

```
switch# clear ipv6 ospfv3 statistics
switch# clear ipv6 ospfv3 statistics interface 1/1/1
switch# clear ipv6 ospfv3 statistics interface 1/1/1 vrf vrf_red
```

dead-interval

Syntax

dead-interval <INTERVAL>

no dead-interval

Description

Sets the interval after which a neighbor is declared dead if no hello packet comes in for virtual links.

The **no** form of this command sets the dead interval to default for virtual links. The default value is 40 seconds (generally four times the hello packet interval).

Command context

config-router-vlink

Parameters

<INTERVAL>

Specifies the time interval for the dead interval, in seconds. Range: 1 to 65535. Default: 40.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 virtual links dead interval:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# dead-interval 30
```

Setting OSPFv3 virtual links dead interval to default:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1  
switch(config-router-vlink6)# no dead-interval
```

default-metric

Syntax

```
default-metric <METRIC-VALUE>
```

```
no default-metric
```

Description

Sets the default metric for redistributed routes in the OSPFv3.

The `no` form of this command sets the default metric to be used for redistributed routes into OSPFv3 to the default of 25.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<METRIC-VALUE>

Specifies the default metric value to use for redistributed routes. Range: 1 to 1677214. Default: 25.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting default metric for redistributed routes:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# default-metric 36
```

Setting default metric for redistributed routes to the default:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no default-metric
```

disable

Syntax

```
disable
```

Description

Disables the OSPFv3 process. By default OSPFv3 process is enabled.



IMPORTANT: This command does not remove the OSPFv3 configurations.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Disabling OSPFv3 process:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# disable
```

distance

Syntax

```
distance <DISTANCE>
```

```
no distance
```

Description

Defines an administrative distance for OSPFv3. Administrative distance is used as a criteria to select the best route when the same route is learned by multiple routing protocols.

The `no` form of this command sets the OSPFv3 administrative distance to the default of 110.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<DISTANCE>

Specifies OSPFv3 administrative distance. Range: 1 to 255. Default: 110.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF administrative distance:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# distance 100
```

Setting OSPF administrative distance to the default:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no distance
```

enable

Syntax

```
enable
```

Description

Enables OSPFv3 process when disabled. By default OSPFv3 process is enabled.

Command context

config-ospfv3-<PROCESS-ID>

Authority

Administrators or local user group members with execution rights for this command.

Example

Enabling OSPFv3 process:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# enable
```

encryption ipsec spi

Syntax

```
encryption ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>  
([<ENCR-TYPE> <KEY-TYPE> <ENCR-KEY>] | null)
```

```
no encryption
```

Description

Configures IPsec ESP for Vlink.

The `no` form of this command removes IPsec ESP for Vlink from the specified area.

Command context

config-router-vlink

Parameters

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec ESP header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. IPsec ESP SPI must be unique on the switch. Range: 256 to 4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: `md5` or `sha1`.

<ENCR-TYPE>

Specifies the algorithm to use for encryption: `des`, `3des` or `aes`.

<KEY-TYPE>

Specifies the key type to use: `plaintext` (unencrypted), `hex-string` (encrypted) or `ciphertext` (encrypted).

<AUTH-KEY>

Specifies the key to use for IPsec.

<ENCR-KEY>

Specifies the encryption key to use for IPsec.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting area 1 to use IPsec ESP for Vlink:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1
switch(config-ospfv3-1)# area 1 virtual-link 3.3.3.3
switch(config-router-vlink6)# encryption ipsec spi 256 md5 plaintext abcd des plaintext abcdefab
```

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1
switch(config-ospfv3-1)# area 1 virtual-link 3.3.3.3
switch(config-router-vlink6)# encryption ipsec spi 256 md5 plaintext abcd null
```

Removing IPsec ESP for Vlink on area 1:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 1 virtual-link 3.3.3.3
switch(config-router-vlink6)# no encryption
```

default-information originate

Syntax

```
default-information originate
```

```
no default-information originate
```

Description

Configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors if it is present in the routing table. The **no** form of this command disables advertisement of the default route.

Command context

```
config-ospf-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting advertisement of the default route:

```
switch(config)# router ospf 1
switch(config-ospf-1)# default-information originate
```

Disabling advertisement of the default route:

```
switch(config)# router ospf 1
switch(config-ospf-1)# no default-information originate
```

default-information originate always

Syntax

```
default-information originate always
```

```
no default-information originate always
```

Description

Configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors, regardless if it is present in the routing table or not.

The `no` form of this command sets the OSPF administrative distance to the default of 110.

Command context

```
config-ospf-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# default-information originate always
```

Disabling advertisement of the default route:

```
switch(config)# router ospf 1  
switch(config-ospf-1)# no default-information originate always
```

graceful-restart

Syntax

```
graceful-restart {restart-interval <INTERVAL> | helper | strict-lsa-check}
```

```
no graceful-restart {restart-interval | helper | strict-lsa-check}
```

Description

Configures graceful restart for OSPFv3. By default graceful restart is enabled on the OSPFv3 router.

The `no` form of this command sets the restart interval to the default of 120 seconds or disables helper mode depending on the specified parameters.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

restart-interval <INTERVAL>

Specifies the time another router waits for this router to gracefully restart and selects the maximum time to wait in seconds. Range: 5 to 1800. Default: 120.

helper

Specifies that the router will participate in the graceful restart of a neighbor router.

strict-lsa-check

Enable strict LSA checking when acting as a restart helper for a restarting peer.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling OSPF graceful restart:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# graceful-restart restart-interval 40
switch(config-ospfv3-1)# graceful-restart helper
switch(config-ospfv3-1)# graceful-restart strict-lsa-check
```

Setting the restart interval to default, disabling helper mode, and strict LSA checking:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no graceful-restart restart-interval
switch(config-ospfv3-1)# no graceful-restart helper
switch(config-ospfv3-1)# no graceful-restart strict-lsa-check
```

hello-interval

Syntax

hello-interval <INTERVAL>

no hello-interval



NOTE: For proper operation, the hello interval must be shorter than the dead interval.

Description

Sets the time interval between OSPF hello packets for virtual links.

The **no** form of this command sets the hello interval to the default value of 10 seconds for virtual links.

Command context

config-router-vlink

Parameters

<INTERVAL>

Specifies the time interval for the hello interval, in seconds. Range: 1 to 65535. Default: 10.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF virtual links hello interval:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# hello-interval 30
```

Setting OSPF virtual links hello interval to default:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# no hello-interval
```


ipv6 ospfv3 area

Syntax

```
ipv6 ospfv3 <PROCESS-ID> area <AREA-ID>
```

```
no ipv6 ospfv3 <PROCESS-ID> area <area-id>
```

Description

Runs the OSPFv3 protocol on the interface for the area specified.

To move an interface to a new area, unmap the existing area and then associate a new area with the interface.

The `no` form of this command disables OSPF on the interface and removes the interface from the area. Interfaces which have an IP address configured on the network or in a subset of the network, stop participating in the OSPF protocol

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<PROCESS-ID>

Specifies the OSPFv3 process ID. Range: 1 to 63.

<AREA-ID>

Specifies the area ID in one of the following formats.

- Area ID in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.
- Area ID as a decimal value. Range: 0-4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 network for the area:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 1 area 1
switch(config-if-vlan)# ipv6 ospfv3 1 area 0.0.0.1
```

Disabling OSPFv3 network for the area:

```
switch(config)# interface 1/1/1
switch(config-if-vlan)# no ipv6 ospfv3 1 area 1
switch(config-if-vlan)# no ipv6 ospfv3 1 area 0.0.0.1
```

ipv6 ospfv3 authentication null

Syntax

```
ipv6 ospfv3 authentication null
```

Description

Configures null authentication on an interface which disables IPsec authentication.

Command context

```
config-if  
config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling IPsec on interface VLAN 1:

```
switch(config)# interface van 1  
switch(config-if-vlan)# ipv6 ospfv3 authentication null
```

ipv6 ospfv3 authentication ipsec

Syntax

```
ipv6 ospfv3 authentication ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>  
  
no ipv6 ospfv3 authentication
```

Description

Configures IPSec AH authentication. OSPFv3 interfaces that have IPsec configured at the interface context will not use area level IPsec.

The `no` form of this command removes IPsec AH authentication for the specified area.

Command context

```
config-if  
config-if-vlan
```

Parameters

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec AH header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. The SPI must be unique on the switch. Range: 256-4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: `md5` or `sha1`.

<KEY-TYPE>

Specifies the key type to use: `plaintext` (not encrypted) or `hex-string` (encrypted) or `ciphertext` (encrypted).

<AUTH-KEY>

Specifies the key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting interface VLAN 1 to use IPsec authentication (AH):

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ipv6 ospfv3 authentication ipsec spi 256 md5 plaintext abcd
```

Removing IPsec authentication (AH) on interface VLAN 1:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 authentication
```

ipv6 ospfv3 cost

Syntax

```
ipv6 ospfv3 cost <INTERFACE-COST>
```

```
no ipv6 ospfv3 cost
```

Description

Sets the cost (metric) associated with a particular interface. The interface cost is used as a parameter to calculate the best routes.

The `no` form of this command sets the cost (metric) associated with a particular interface to the default of 1.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<INTERFACE-COST>

Specifies the interface cost value. Range: 1 to 65535. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 interface cost:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ipv6 ospfv3 cost 100
```

Setting the OSPFv3 interface cost to default:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 cost
```

ipv6 ospfv3 dead-interval

Syntax

```
ipv6 ospfv3 dead-interval <INTERVAL>
```

```
no ipv6 ospfv3 dead-interval
```

Description

Sets the interval after a neighbor is declared dead when no hello packet is received on the OSPFv3 interface. The `no` form of this command sets the interval after which a neighbor is declared dead, to the default for the OSPFv3 interface. The default value is 40 seconds (generally four times the hello packet interval).

Command context

```
config-if
config-if-vlan
```

Parameters

<INTERVAL>

Specifies the time interval for the dead interval, in seconds. Range: 1 to 65535. Default: 40.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 dead interval on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 dead-interval 30
```

Setting OSPFv3 dead interval to default on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ipv6 ospfv3 dead-interval
```

ipv6 ospfv3 encryption ipsec

Syntax

```
ipv6 ospfv3 encryption ipsec spi <SPI-INDEX> <AUTH-TYPE> <KEY-TYPE> <AUTH-KEY>
({<ENCR-TYPE> <KEY-TYPE> <ENCR-KEY>} | null)
```

```
no ipv6 ospfv3 encryption
```

Description

Configures IPsec ESP. OSPFv3 interfaces that have IPsec configured at the interface context will not use area level IPsec ESP.

The `no` form of this command removes IPsec ESP for the specified area.

Command context

```
config-if
config-if-vlan
```

Parameters

spi <SPI-INDEX>

Specifies the Security Parameters Index (SPI) to use. The SPI is an identification tag carried in the IPsec ESP header. It enables the receiving OSPF process to select and use the Security Association (SA) from the SA table. The SPI must be unique on the switch. Range: 256 to 4294967295 characters.

<AUTH-TYPE>

Specifies the algorithm to use for authentication: md5 or sha1.

<ENCR-TYPE>

Specifies the algorithm to use for encryption: des, 3des or aes.

<KEY-TYPE>

Specifies the key type to use: plaintext (not encrypted), hex-string (encrypted) or ciphertext (encrypted).

<AUTH-KEY>

Specifies the authentication key.

<ENCR-KEY>

Specifies the encryption key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting interface VLAN 1 to use IPsec ESP:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 encryption ipsec spi 256 sha1 plaintext abcdef aes
plaintext abcdefabcdefabcd
```

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 encryption ipsec spi 256 sha1 plaintext abcdef null
```

Removing IPsec on interface VLAN 1:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ipv6 ospfv3 encryption
```

ipv6 ospfv3 encryption null

Syntax

```
ipv6 ospfv3 encryption null
```

Description

Configures NULL ESP on an interface which disables IPsec ESP.

Command context

```
config-if
config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disable IPsec ESP on interface VLAN 1:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ipv6 ospfv3 encryption null
```

ipv6 ospfv3 hello-interval

Syntax

```
ipv6 ospfv3 hello-interval <INTERVAL>
```

```
no ipv6 ospfv3 hello-interval
```

Description

Sets the time interval between OSPFv3 hello packets for the OSPFv3 interface.

The `no` form of this command sets the time interval between OSPFv3 hello packets to the default for the OSPFv3 interface of 10 seconds.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<INTERVAL>

Specifies the time interval between hello packets, in seconds. Range: 1 to 65535. Default: 10.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 hello interval on the interface:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ipv6 ospfv3 hello-interval 30
```

Setting OSPFv3 hello interval to default on the interface:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 hello-interval
```

ipv6 ospfv3 network

Syntax

```
ipv6 ospfv3 network {broadcast|point-to-point}
```

```
no ipv6 ospfv3 network
```

Description

Configures the network type for the interface. By default the network type is broadcast network.

The `no` form of this command sets the network type for the interface to the system default of broadcast network.

Command context

config-if
config-if-vlan

Parameters

broadcast

Specifies the OSPFv3 network type as a broadcast multiaccess network.

point-to-point

Specifies the OSPFv3 network type as a point-to-point network.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 network type for the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 network broadcast
switch(config-if-vlan)# ipv6 ospfv3 network point-to-point
```

Disabling OSPFv3 network type for the interface to system default of broadcast network:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ipv6 ospfv3 network
```

ipv6 ospfv3 passive

Syntax

ipv6 ospfv3 passive

no ipv6 ospfv3 passive

Description

Configures the interface as an OSPFv3 passive interface. With this setting, the interface participates in the OSPF, but does not send or receive OSPF packets on that interface.

The **no** form of this command resets the interface as active. With this setting, the interface starts sending and receiving OSPF packets.

Command context

config-if
config-if-vlan

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the interface as OSPFv3 passive interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 passive
```

Setting the interface as OSPFv3 active interface:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 passive
```

ipv6 ospfv3 priority

Syntax

```
ipv6 ospfv3 priority <number-value>
```

```
no ipv6 ospfv3 priority
```

Description

Sets the OSPFv3 priority for the interface. The larger the numeric value of the priority, the higher the chance it will become the designated router. Setting a priority of 0 makes the router ineligible to become a designated router or back up designated router.

The `no` form of this command sets the OSPFv3 priority for the interface to the default of 1.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<number-value>

Specifies the OSPFv3 priority value. Default: 1. Range: 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the OSPFv3 priority for the interface:

```
switch(config)# interface vlan /1  
switch(config-if-vlan)# ipv6 ospfv3 priority 50
```

Setting the OSPFv3 priority for the interface to the default of 1:

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 priority
```

ipv6 ospfv3 retransmit-interval

Syntax

```
ipv6 ospfv3 retransmit-interval <INTERVAL>
```

```
no ipv6 ospfv3 retransmit-interval
```

Description

Sets the time between retransmitting lost link state advertisements for the OSPFv3 interface.

The `no` form of this command sets the time between retransmitting lost link state advertisements to the default 5 seconds.

Command context

config-if
config-if-vlan

Parameters

<INTERVAL>

Specifies the time interval for the retransmit interval, in seconds. Range: 1 to 3600. Default: 5

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 retransmit interval on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 retransmit-interval 30
```

Setting OSPFv3 retransmit interval to the default on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ipv6 ospfv3 retransmit-interval
```

ipv6 ospfv3 shutdown

Syntax

ipv6 ospfv3 shutdown

no ipv6 ospfv3 shutdown

Description

Disables OSPFv3 on the interface. The interface state changes to Down. It does not remove the interface from the OSPF area. To remove the interface, use the command `no ip ospf area`.

The `no` form of this command re-enables OSPFv3 on the interface.

Command context

config-if
config-if-vlan

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling OSPFv3 on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# ipv6 ospfv3 shutdown
```

Re-enabling OSPFv3 on the interface:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no ipv6 ospfv3 shutdown
```

ipv6 ospfv3 transit-delay

Syntax

```
ipv6 ospfv3 transit-delay <DELAY>
```

```
no ipv6 ospfv3 transit-delay
```

Description

Sets the time delay in Link state transmission for the OSPFv3 interface.

The `no` form of this command sets the transit delay in Link state transmission to the default of 1 second.

Command context

```
config-if
```

```
config-if-vlan
```

Parameters

<DELAY>

Specifies the time delay for the transit delay, in seconds. Range: 1 to 3600. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 transit delay on the interface

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# ipv6 ospfv3 transit-delay 30
```

Setting OSPFv3 transit delay to default on the interface

```
switch(config)# interface vlan 1  
switch(config-if-vlan)# no ipv6 ospfv3 transit-delay
```

maximum-paths

Syntax

```
maximum-paths <MAXIMUM>
```

```
no maximum-paths
```

Description

Sets the maximum number of ECMP routes that OSPFv3 can support.

The `no` form of this command sets the maximum number of ECMP routes that OSPFv3 can support to the default value of 4.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<MAXIMUM>

Specifies the maximum number of ECMP routes. Range: 1 to 8. Default: 4.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting maximum number of parallel routes:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# maximum-paths 3
```

Setting maximum number of parallel paths to the default value of 4:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no maximum-paths
```

max-metric router-lsa

Syntax

```
max-metric router-lsa [on-startup <INTERVAL>]
```

```
no max-metric router-lsa [on-startup]
```

Description

Sets the protocol to advertise a maximum metric so that other routers do not prefer the router as an intermediate hop in their shortest path first (SPF) calculations. If the on-startup parameter is used, the router is configured to advertise a maximum metric at startup. That is, for the time specified in seconds, or the default value of 600 seconds.

To disable advertisement of the maximum metric, use the `no` form of the command.

The `no` form of this command advertises the normal cost metrics instead of advertising the maximized cost metric. This setting causes the router to be considered in traffic forwarding.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

on-startup <INTERVAL>

Automatically advertises the stub Router-LSA (or maximize the router-LSA cost metric) for a specified time interval upon OSPFv3 startup.

Specifies the time in seconds. Range: 5 to 86400. Default: 600.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting to maximize the cost metrics for Router LSA:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# max-metric router-lsa  
switch(config-ospfv3-1)# max-metric router-lsa on-startup 3000
```

Setting to advertise the normal cost metrics instead of advertising the maximized cost metric:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no max-metric router-lsa  
switch(config-ospfv3-1)# no max-metric router-lsa on-startup
```

passive-interface default

Syntax

```
passive-interface default
```

```
no passive-interface default
```

Description

Sets all OSPFv3 interfaces as passive.

The `no` form of this command sets all the OSPFv3 interfaces as active.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3-enabled interfaces as passive:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# passive-interface default
```

Setting OSPFv3-enabled interfaces as active:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no passive-interface default
```

redistribute

Syntax

```
redistribute {connected | static | ripng | ospfv3 <PROCESS-ID>} [route-map <ROUTE-MAP-NAME>]
```

```
no redistribute {connected | static} [route-map <ROUTE-MAP-NAME>]
```

Description

Redistributes routes originating from other protocols, or from another OSPF process, to the current OSPF process.

The `no` form of this command disables redistribution of routes to the current OSPF process.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

connected

Specifies redistributing connected routes (directly attached subnet or host) to OSPFv3.

static

Specifies redistributing static routes to OSPFv3.

ripng

Specifies redistributing RIPng routes.

ospfv3 <PROCESS-ID>

Specifies redistributing routes from the specified OSPFv3 process ID.

route-map <ROUTE-MAP-NAME>

Specifies the name of a route map. To create a route map, use the `route-map` command.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Redistributing routes to OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# redistribute bgp
switch(config-ospfv3-1)# redistribute connected
switch(config-ospfv3-1)# redistribute static
switch(config-ospfv3-1)# redistribute ripng
switch(config-ospfv3-1)# redistribute ospfv3 1
switch(config-ospfv3-1)# redistribute route-map static_networks
```

Disabling redistributing routes to OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no redistribute bgp
switch(config-ospfv3-1)# no redistribute connected
switch(config-ospfv3-1)# no redistribute static
switch(config-ospfv3-1)# no redistribute ripng
switch(config-ospfv3-1)# no redistribute ospfv3 1
switch(config-ospfv3-1)# no redistribute route-map static_networks
```

reference-bandwidth

Syntax

```
reference-bandwidth <BANDWIDTH>
```

```
no reference-bandwidth
```

Description

Sets the reference bandwidth for OSPFv3. If the OSPFv3 interface cost is not explicitly set, then the cost of all the OSPFv3 interfaces is recalculated based on the reference bandwidth and link speed of the interface.

For VLAN interfaces the calculated link speed value is 1 Gbps (if the OSPFv3 interface cost is not explicitly set).

The `no` form of this command sets the reference bandwidth for OSPF to the default of 100000 Mbps.

Command context

config-ospfv3-<PROCESS-ID>

Parameters

<BANDWIDTH>

Specifies the reference bandwidth used to calculate the cost of an interface in Mbps. Range: 1 to 4000000. Default: 100000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the reference bandwidth:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# reference-bandwidth 40000
```

Setting the reference bandwidth to the default value:

```
switch(config)# routerv3 ospf 1
switch(config-ospfv3-1)# no reference-bandwidth
```

retransmit-interval

Syntax

retransmit-interval <INTERVAL>

no retransmit-interval

Description

Sets the time between retransmitting lost link state advertisements for virtual links.

The **no** form of this command sets the time between retransmitting lost link state advertisements to the default of 5 seconds for virtual links.

Command context

config-router-vlink

Parameters

<INTERVAL>

Specifies the time interval for the retransmit interval, in seconds. Range: 1 to 3600. Default: 5.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 virtual links retransmit interval:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# retransmit-interval 30
```

Setting OSPFv3 virtual links retransmit interval to default:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1  
switch(config-router-vlink6)# no retransmit-interval
```

router-id

Syntax

```
router-id <ROUTER-ADDRESS>
```

```
no router-id
```

Description

Sets an ID for the router in an IPv4 address format.

The `no` form of this command unconfigures the router-id for the instance and sets the router-id to the default. The router-id is changed to the dynamically selected router-id. The default router-id 0.0.0.0 updates to the routing stack that triggers to auto-elect a router-id based on the highest IP address of loopback interface, or the highest IP address of interfaces.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<ROUTER-ADDRESS>

Specifies the router address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting router-id in the OSPFv3 context:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1) # router-id 1.1.1.1
```

Unconfiguring router-id:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1) # no router-id
```

router ospfv3

Syntax

```
router ospfv3 <PROCESS-ID> [vrf <VRF-NAME>]
```

```
no router ospfv3 <PROCESS-ID> [vrf <VRF-NAME>]
```

Description

Creates the OSPFv3 process (if not created already) and enters the router OSPFv3 instance context. Optionally if specified, you can specify a named VRF, or the default VRF if the `<vrf-name>` is not specified. Only one OSPFv3 process is allowed per VRF.

The **no** form of this command removes the OSPFv3 instance. If a VRF is specified, it removes the OSPF instance from the named VRF, or the default VRF if the *<var-name>* is not specified.

Command context

config

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID. Length: 1 to 63.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Entering the router OSPFv3 instance:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)#
```

Setting the router OSPFv3 VRF instance:

```
switch(config)# router ospfv3 1 vrf vrf_red
```

Removing the router OSPFv3 instance:

```
switch#(config)# no router ospfv3 1
```

show ipv6 ospfv3

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows OSPFv3 information including area, state, and configuration information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID optionally to show OSPFv3 information for a particular OSPFv3 process. Range: 1 to 63.

all-vrfs

Select to show OSPFv3 information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Showing general OSPFv3 configurations:

```
switch# show ipv6 ospfv3 1
Routing Process 1 with ID: 1.1.1.1 VRF default
-----

OSPFv3 Protocol is enabled
Graceful-restart is configured
Restart Interval: 120, State: inactive
Last Graceful Restart Exit Status: none
Area Border: false
AS Border: false
SPF: Start Time: 200ms, Hold Time: 1000ms, Max Wait Time: 5000ms
LSA: Start Time: 5000ms, Hold Time: 0ms, Max Wait Time: 0ms
LSA Arrival: 1000ms
Maximum Paths to Destination: 4
Number of External LSAs 0, checksum sum 0
Summary address:
  prefix fd00::0/64 advertise tag 10
Number of areas is: 2, 2 normal, 0 stub, 0 NSSA
Number of active areas is: 1, 1 normal, 0 stub, 0 NSSA
Reference Bandwidth: 100000 Mbps
Area (0.0.0.0) (Active)
  Interfaces in this Area: 1 Active Interfaces: 1
  Passive Interfaces: 0 Loopback Interfaces: 0
  SPF calculation has run 9 times
  Number of LSAs: 4, checksum sum 195745
Area (0.0.0.1) (Inactive)
  Interfaces in this Area: 0 Active Interfaces: 0
  Passive Interfaces: 0 Loopback Interfaces: 0
  SPF calculation has run 8 times
Area ranges:
  fd00::1/64, inter-area, advertise
  Number of LSAs: 0, checksum sum 0
```

show ipv6 ospfv3 border-routers

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] border-routers [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows the OSPFv3 routing table entries for Area Border Router (ABR) and Autonomous System Border Router (ASBR).

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID to show the OSPFv3 routing table entries for ABR and ASBR for the particular OSPFv3 process. Range: 1-63.

all-vrfs

Select to show OSPFv3 border router information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

On the 6400 Switch Series, interface identification differs.

Showing OSPFv3 border routers information for default VRF:

```
switch# show ipv6 ospfv3 border-routers
OSPFv3 Process ID 1 VRF default, Internal Routing Table
-----
Codes: i - Intra-area route, I - Inter-area route
i 2.2.2.2 [10], ABR, Area 0.0.0.0, SPF 17 via
    fe80::98f2:b301:2c68:1d48, Interface 1/1/1
i 2.2.2.2 [10], ABR, Area 0.0.0.1, SPF 17 via
    fe80::98f2:b301:3068:1d48, Interface 1/1/2
```

show ipv6 ospfv3 interface

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] interface [<INTERFACE-NAME>] [brief]
[all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows information about OSPFv3 enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID optionally to show the OSPFv3 enabled interfaces for the particular OSPFv3 process. Range: 1-63.

<INTERFACE-NAME>

Selects to show information only for the specified OSPFv3 enabled interface.

brief

Select to show the overview information for the OSPFv3 enabled interface.

all-vrfs

Select to show OSPF-enabled interface information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

On the 6400 Switch Series, interface identification differs.

Showing OSPFv3 information for all interfaces in default VRF:

```
switch# show ipv6 ospfv3 interface
Interface 1/1/1 is up, Line Protocol is up
-----
IPv6 address fe80::98f2:b301:468:be02
Process ID 1 VRF default, Area 0.0.0.0
  State Dr, Status up, Network type Broadcast
  Link Speed: 10000 Mbps
  Cost Configured NA, Calculated 10
  Transit Delay 1 sec, Router Priority 1
  Designated Router ID: 1.1.1.1
  Backup Designated Router ID: 2.2.2.2
  Timer Intervals: Hello 10, Dead 40, Retransmit 5
  Number of Link LSAs: 2, checksum sum 67473

Interface 1/1/2 is up, Line Protocol is up
-----
IPv6 address fe80::98f2:b301:868:be02
Process ID 1 VRF default, Area 0.0.0.1
  State Waiting, Status up, Network type Broadcast
  Link Speed: 10000 Mbps
  Cost Configured NA, Calculated 10
  Transit Delay 1 sec, Router Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer Intervals: Hello 10, Dead 40, Retransmit 5
  Number of Link LSAs: 1, checksum sum 24592
```

Showing overview information for OSPFv3 enabled interfaces for all VRFs:

```
switch# show ipv6 ospfv3 interface brief all-vrfs
OSPFv3 Process ID 1 VRF default
=====

Total Number of Interfaces: 2

Interface      Area          Cost          State          Status
```

1/1/1	0.0.0.0	10	Dr	up
1/1/2	0.0.0.1	10	Backup-dr	up

show ipv6 ospfv3 neighbors

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] neighbors [<NEIGHBOR-ID>]
    [interface <INTERFACE-NAME>] [detail | summary]
    [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows information about OSPFv3 neighbors.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID to show OSPFv3 neighbor information for the particular OSPFv3 process.
Range: 1-63.

neighbors <NEIGHBOR-ID>

Shows information about a particular neighbor, specified in IPv4 format (A.B.C.D).

interface <INTERFACE-NAME>

Shows neighbor information only for the specified interface.

detail

Shows detailed information for the neighbors.

summary

Shows summary information for the neighbors.

all-vrfs

Shows neighbor information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

On the 6400 Switch Series, interface identification differs.

Showing OSPFv3 neighbors information:

```
switch# show ipv6 ospfv3 neighbors
```

```
OSPFv3 Process ID 1 VRF default
```

```
=====
```

```
Total Number of Neighbors: 1
```

Neighbor ID	Priority	State	Interface
2.2.2.2	1	FULL/BDR	1/1/1
Neighbor address fe80::98f2:b301:2c68:1d48			

Showing OSPFv3 neighbors information for a specific neighbor:

```
switch# show ipv6 ospfv3 neighbors 2.2.2.2
```

```
Neighbor 2.2.2.2, Interface address fe80::98f2:b301:2c68:1d48
```

```
-----
```

```
Process ID 1 VRF default, in Area 0.0.0.0 via Interface 1/1/1
```

```
Neighbor priority is 1, State is FULL
```

```
Options is 0x13
```

```
Dead Timer due in 00:00:38
```

```
Time since last state change 00h:20m:26s
```

Showing detail OSPFv3 neighbors information for a specific neighbor:

```
switch# show ipv6 ospfv3 neighbors 2.2.2.2 detail
```

```
Neighbor 2.2.2.2, Interface address fe80::98f2:b301:2c68:1d48
```

```
-----
```

```
Process ID 1 VRF default, in Area 0.0.0.0 via Interface 1/1/1
```

```
Neighbor priority is 1, State is FULL
```

```
DR is 1.1.1.1, BDR is 2.2.2.2
```

```
Options is 0x13
```

```
Dead Timer due in 00:00:36
```

```
Retransmission Queue Length 0
```

```
Time since last state change 00h:20m:38s
```

Showing OSPFv3 neighbors information for interface 1/1/1:

```
switch# show ipv6 ospfv3 neighbors 2.2.2.2 interface 1/1/1
```

```
Neighbor 2.2.2.2, Interface address fe80::98f2:b301:2c68:1d48
```

```
-----
```

```
Process ID 1 VRF default, in Area 0.0.0.0 via Interface 1/1/1
```

```
Neighbor priority is 1, State is FULL
```

```
Options is 0x13
```

```
Dead Timer due in 00:00:32
```

```
Time since last state change 00h:20m:52s
```

Showing summary OSPFv3 neighbors information for a specific neighbor for all VRFs:

```
switch# show ipv6 ospfv3 neighbors 3.3.3.3 summary all-vrfs
```

```
Interface  Down Attempt Init TwoWay ExStart Exchange Loading Full Total
```

```
-----
```

```
1/1/1      0    0      0    0      0      0      0      1    1
```

```
1/1/2      0    0      0    0      0      0      0      0    0
```

```
Total     0    0      0    0      0      0      0      1    1
```

Showing OSPFv3 neighbors information for VRF red:

```
switch# show ipv6 ospfv3 neighbors vrf red
```

```
OSPFv3 Process ID 1 VRF red
```

```
=====
```

```
Total Number of Neighbors: 1
```

Neighbor ID	Priority	State	Interface
-------------	----------	-------	-----------

```
-----
1.1.1.1          1          FULL/BDR          1/1/1
Neighbor address fe80::800:901:42c:7ddd
```

show ipv6 ospfv3 routes

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] routes [<PREFIX/LENGTH>]
               [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows the OSPFv3 routing table information.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID that shows information from the OSPFv3 routing table for the particular OSPFv3 process. Range: 1-63.

<PREFIX/LENGTH>

Specifies the IPv6 destination prefix showing information about a particular destination prefix. For example, 2010:bd9::/32.

all-vrfs

Select to show OSPFv3 routing table information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

On the 6400 Switch Series, interface identification differs.

Showing OSPFv3 routing table information:

```
switch# show ipv6 ospfv3 routes
Codes: i - Intra-area route, I - Inter-area route
       E1 - External type-1, E2 - External type-2

OSPFv3 Process ID 1 VRF default, Routing Table
-----

Total Number of OSPFv3 Routes : 2

111::/64          (i) area:0.0.0.0
```

```
    directly attached to interface 1/1/1, cost 1 distance 110
fd00::/64          (i) area:0.0.0.1
    directly attached to interface vlan10, cost 1 distance 110
```

Showing OSPFv3 routing table information for VRF red:

```
switch# show ipv6 ospfv3 routes vrf red

Codes: i - Intra-area route, I - Inter-area route
       E1 - External type-1, E2 - External type-2

OSPFv3 Process ID 2 VRF red, Routing Table
-----

Total Number of OSPFv3 Routes : 1

222::/64          (i) area:0.0.0.1
    directly attached to interface 1/1/2, cost 1 distance 110
```

Showing OSPFv3 routing table information for destination prefix fd00::/64:

```
switch# show ipv6 ospfv3 1 routes fd00::/64

Codes: i - Intra-area route, I - Inter-area route
       E1 - External type-1, E2 - External type-2

OSPFv3 Process ID 1 VRF default, Routing Table for prefixes fd00::/64
-----

Total Number of OSPFv3 Routes : 1

fd00::/64          (i) area:0.0.0.1
    directly attached to interface vlan10, cost 1 distance 110
```

show ipv6 ospfv3 statistics

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] statistics
                [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows OSPFv3 statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID that shows information on the OSPFv3 SPF statistics for the particular OSPFv3 process. Range: 1-63.

all-vrfs

Select to show OSPFv3 SPF statistics information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing OSPFv3 statistics information:

```
switch# show ipv6 ospfv3 statistics
OSPFv3 Process ID 1 VRF default, Statistics (cleared 3h 2m 21s ago)
-----
Unknown Interface Drops           : 0
Unknown Virtual Interface Drops   : 0
Bad IPv6 Header Length Drops      : 0
Wrong OSPFv3 Version Drops        : 0
Bad Source IPv6 Drops             : 0
Resource Failure Drops            : 0
Bad Header Length Drops           : 0
Total Drops                       : 0
```

Showing OSPFv3 statistics information for VRF red:

```
switch# show ipv6 ospfv3 2 statistics vrf red
OSPFv3 Process ID 2 VRF red, Statistics (cleared 3h 2m 30s ago)
-----
Unknown Interface Drops           : 0
Unknown Virtual Interface Drops   : 0
Bad IPv6 Header Length Drops      : 0
Wrong OSPFv3 Version Drops        : 0
Bad Source IPv6 Drops             : 0
Resource Failure Drops            : 0
Bad Header Length Drops           : 0
Total Drops                       : 0
```

show ipv6 ospfv3 statistics interface

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] statistics interface [<INTERFACE-NAME>]
                [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows the OSPFv3 statistics for the OSPFv3-enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies an OSPFv3 process ID to show OSPF-enabled interface statistics information on the specified OSPFv3 process. Range: 1 to 63.

<INTERFACE-NAME>

Selects to show information only for the specified interface.

all-vrfs

Select to show OSPF-enabled interface statistics information for all VRFs.

vrf <VRF-NAME>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Showing OSPFv3-enabled interfaces information for interface 1/1/1:

```
switch# show ipv6 ospfv3 statistics interface all-vrfs
OSPFv3 Process ID 1 VRF default, Interface vlan2000 Statistics (cleared 0h 2m 52s ago)
=====
Tx Hello packets      : 0          Rx Hello packets      : 0
Tx Hello bytes        : 0          Rx Hello bytes        : 0
Tx DD packets         : 0          Rx DD packets         : 0
Tx DD bytes           : 0          Rx DD bytes           : 0
Tx LS request packets : 0          Rx LS request packets : 0
Tx LS request bytes   : 0          Rx LS request bytes   : 0
Tx LS update packets  : 0          Rx LS update packets  : 0
Tx LS update bytes    : 0          Rx LS update bytes    : 0
Tx LS ack packets     : 0          Rx LS ack packets     : 0
Tx LS ack bytes       : 0          Rx LS ack bytes       : 0

Total IPsec packets processed : 0
Total IPsec bytes processed   : 0
Total Number of State Changes : 0
Number of LSAs                : 0
LSA Checksum Sum              : 0

Total OSPFv3 Packets Discarded: 0
-----
Reason                        Packets Dropped
-----
Invalid Type                  0
Invalid Length                0
Invalid Version               0
Bad or Unknown Source         0
Area Mismatch                 0
Self-originated               0
Duplicate Router ID           0
Interface Standby             0
```

```

Total Hello Packets Dropped      0
  Hello Interval Mismatch        0
  Dead Interval Mismatch         0
  Options Mismatch               0
  MTU Mismatch                   0
  Neighbor Ignored               0
Resource Failures                0
Bad LSA Length                   0
Others                           0
IPsec Authentication Errors      0
IPsec ESP Errors                 0

Total LSAs Ignored : 0
-----
Bad Type           : 0
Bad Length         : 0
Invalid Data       : 0
Invalid Checksum   : 0

```

show ipv6 ospfv3 virtual-links

Syntax

```
show ipv6 ospfv3 [<PROCESS-ID>] virtual-links [brief]
[all-vrfs | vrf <vrf-name>] [vsx-peer]
```

Description

Displays the current state and parameters of the OSPFv3 virtual links.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Enter an OSPFv3 process ID to display information on the OSPFv3 virtual links for the particular OSPFv3 process. Range: 1 to 63.

brief

Select to display brief overview information for the OSPFv3 virtual links.

all-vrfs

Select to display OSPFv3 virtual links information for all VRFs.

vrf <vrf-name>

Specifies the name of a VRF. Default: default.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Show OSPFv3 virtual links information:

```
switch# show ipv6 ospfv3 virtual-links
Virtual link to router 4.4.4.4 is down
-----

Process ID 1 VRF default, Transit area 0.0.0.1
Transit delay 1 sec
Timer Intervals: hello 10, dead 40, retransmit 5
Number of Link LSAs: 0, checksum sum 0
0 state changes
AH Authentication: MD5, SPI: 256
```

Show brief overview information for OSPFv3 virtual links:

```
switch# show ospfv3 virtual-links brief
OSPFv3 Process ID 1 VRF default
=====

Total Number of Virtual Links: 1

Remote Router      Transit Area      Status
-----
4.4.4.4            0.0.0.1           down
```

summary-address

Syntax

summary-address <IPV6-ADDR>/<MASK> [no-advertise | tag <TAG-VALUE>]

no summary-address <prefix/length> [no-advertise | tag <tag-value>]

Description

Summarizes the external routes with the matching address and mask. When advertising this route, its metric is set to the lowest cost path from among the routes that were summarized.

The `no` form of this command disables route summarization.



NOTE: This command only works for an ASBR (Autonomous System Boundary Router).

Command context

config-ospfv3-<PROCESS-ID>

Parameters

<IPV6-ADDR>

Specifies an IP address in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

no-advertise

Do not advertise the aggregate route. Suppress routes that match the specified prefix/mask pair.

tag <TAG-VALUE>

Specify the tag for the aggregate route. The summary prefix will be advertised along with the tag value in External LSAs. Range: 0 to 4294967295

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPF route summarization:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# summary-address 2001:DB8::1/32
```

Disabling route summarization:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no summary-address 2001:DB8::1/32
```

timers lsa-arrival

Syntax

```
timers lsa-arrival <DELAY>
```

```
no timers lsa-arrival
```

Description

Configures the minimum delay between receiving the same LSA from a peer. The same LSA is an LSA that contains the same LSA ID number, LSA type, and advertising router ID. If an instance of the same LSA arrives sooner before the delay expires, the LSA is dropped. Generally, the LSA arrival timer should be set to a value less than or equal to the start-time value for the command `timers throttle lsa start` on the neighbor.

The `no` form of this command sets the LSA timers to default values.

Command context

```
config-ospfv3-<PROCESS-ID>
```

Parameters

<DELAY>

Specifies the delay in milliseconds. Range: 0 to 600000. Default: 1000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the LSA arrival timer:

```
switch(config)# router ospf 1
switch(config-ospfv3-1)# timers lsa-arrival 10
```

Setting the LSA arrival timer to default:

```
switch(config)# router ospf 1
switch(config-ospfv3-1)# no timers lsa-arrival
```

timers throttle lsa

Syntax

```
timers throttle lsa start-time <START-TIME> hold-time <HOLD-TIME> max-wait-time <WAIT-TIME>

no timers throttle lsa
```

Description

Configures the timers for LSA generation.

The `no` form of this command sets the LSA timers to default values.

Command context

config-ospfv3-<PROCESS-ID>

Parameters

start-time <START-TIME>

Specifies the initial wait time in milliseconds after which LSAs are generated. When set to 0, the LSAs are generated without any delay. Range: 0 to 600000. Default: 5000.

hold-time <HOLD-TIME>

Specifies the amount of time, in milliseconds, between regeneration of an LSA. The hold time doubles each time the same LSA must be regenerated, until `max-wait-time` is reached. When set to 0, LSA regeneration time is not increased. Range: 0 to 600000. Default: 0.

max-wait-time <WAIT-TIME>

Specifies the maximum wait time, in milliseconds, for regeneration of the same LSA. When set to 0, LSA regeneration time is not increased. Range: 0 to 600000. Default: 0.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the LSA timers:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# timers throttle lsa start-time 100 hold-time 1000 max-wait-time 10000
```

Setting LSA timers to default values:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no timers throttle lsa
```

timers throttle spf

Syntax

```
timers throttle spf start-time <START-TIME> hold-time <HOLD-TIME>
max-wait-time <WAIT-TIME>
```

```
no timers throttle spf
```

Description

Configures timers for SPF calculation. There are three timers:

- `start-time` Is the initial delay before an SPF calculation is started. Default is 200 milliseconds.
- `hold-time` Is the progressive backoff time to wait before next scheduled SPF calculation. Default is 1000 milliseconds. If a route change event occurs during this period, the value doubles until it reaches the *max-wait-time*.
- `max-wait-time` Is the maximum time to wait before the next scheduled SPF calculation. Default is 5000 milliseconds. This is used to limit the SPF hold timer and also defines the time to be considered for which the OSPF LSDB has to be stable, after which the SPF throttle mechanism is reset.

The `no` form of this command sets all the configured non-default timers to default value.

Command context

```
config-ospf-<PROCESS-ID>
```

Parameters

<START-TIME>

Time in milliseconds to set timer for initial SPF delay. Default: 200.

<HOLD-TIME>

Time in milliseconds to set the minimum hold time between two consecutive SPF calculations. Default: 1000.

<WAIT-TIME>

Time in milliseconds to set the maximum wait time between two consecutive SPF calculations. Default: 5000.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting non-default timer values for SPF throttling:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# timers throttling spf start-time 500 hold-time 3000 max-wait-time 9000
Switch(config-ospfv3-1)# show running-config current-context
router ospfv3 1
  area 0.0.0.0
  area 0.0.0.1
  area 0.0.0.2 nssa no-summary
  area 0.0.0.3 stub
```

Setting default timer values for SPF throttling after configuring non-default values:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# no timers throttling spf
```

transit-delay

Syntax

```
transit-delay <DELAY>
```

```
no transit-delay
```

Description

Sets the time delay in Link state transmission for virtual links.

The `no` form of this command sets the delay in Link state transmission to the default of 1 second for virtual links.

Command context

```
config-router-vlink
```

Parameters

<DELAY>

Specifies the time delay for the transit delay, in seconds. Range: 1 to 3600. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting OSPFv3 virtual links transit delay:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# transit-delay 30
```

Setting OSPFv3 virtual links transit delay to default:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# area 100 virtual-link 100.0.1.1
switch(config-router-vlink6)# no transit-delay
```

trap-enable

Syntax

```
trap-enable
```

```
no trap-enable
```

Description

Enables the notification of the events to be sent as traps to the SNMP management stations for OSPFv3.

The `no` form of this command disables the notification of the events to be sent as traps to the SNMP management stations for OSPFv3.

Command context

```
config-ospfv3
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling sending notification of events as traps:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# trap-enable
```

Disabling sending notification of events as traps:

```
switch(config)# router ospfv3 1  
switch(config-ospfv3-1)# no trap-enable
```


Overview

BGPv4 (RFC 4271) is an interdomain path-vector routing protocol that runs over TCP on port 179. It provides a reliable loop-free routing between different Autonomous Systems (ASs). BGP routers (called speakers) form peerings (also known as neighbor relationships) with other BGP peers. These peerings can either be internal (iBGP: within the same AS) or external, (eBGP: connecting two different ASs).

The characteristics of BGP are:

- Controls route propagation and the selection of optimal routes, rather than route discovery and calculation. This makes BGP different from interior gateway protocols (IGP) such as OSPF and RIP.
- Uses TCP to enhance reliability.
- Supports CIDR.
- Reduces bandwidth consumption by advertising only incremental updates, which allows advertising large amounts of routing information on the Internet.
- Eliminates routing loops by adding AS path information to BGP routes.
- Provides policies to implement flexible route filtering and selection.
- Provides scalability.

BGP views an autonomous system as a collection of routes under the control of a single organization, using one or more IGPs to route packets within the AS. The IGP could also be an iBGP within the AS and could use BGP as the only routing protocol.

Autonomous system numbers

Organizations requiring connectivity to the Internet must obtain an Autonomous System Number (ASN). ASNs were originally 2 bytes (16 bit), providing 65,535 ASNs. ASNs 64,512-65,534 are private ASNs within the 16-bit ASN range.

BGP sessions

A BGP session refers to the established adjacency between two BGP routers. BGP sessions are always point-to-point and are categorized into two types:

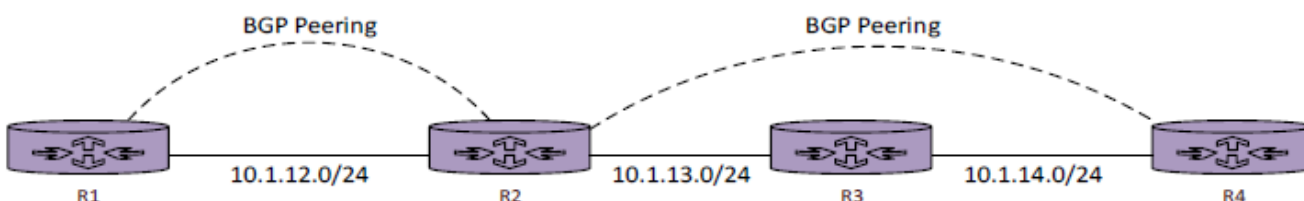
- **Internal BGP (iBGP):** Sessions established with an iBGP router that are in the same AS. iBGP prefixes are assigned an administrative distance (AD) of 200 upon installing into the router routing information base (RIB).
- **External BGP (eBGP):** Sessions established with a BGP router that are in a different AS. eBGP prefixes are assigned an AD of 20 upon installing into the router RIB.

Inter-router communication

BGP does not use hello packets to discover neighbors like IGP protocols and cannot discover neighbors dynamically. BGP was designed as an inter-autonomous routing protocol, implying neighbor adjacencies will not change frequently and are manually configured. BGP neighbors are defined by an IP address.

BGP uses TCP port 179 to communicate with other routers. While BGP can form neighbor adjacencies that are directly connected, it can also form adjacencies that are multiple hops away. Multihop sessions require that the router use an underlying route installed in RIB to establish the TCP session with a remote endpoint.

In the following scenario, R1 is able to establish a direct BGP session with R2. In addition, R2 is able to form a BGP session with R4, even though it passes through R3.



BGP messages

BGP communication uses four message types, as shown in the following table:

Type	Name	Functional Overview
1	OPEN	Sets up and establishes BGP adjacency
2	UPDATE	Advertises, updates, or withdraw routes
3	NOTIFICATION	Indicates and error condition to a BGP neighbor
4	KEEPALIVE	Ensures that BGP neighbors are still alive

BGP neighbor states

BGP forms a TCP session with neighbor routers called peers. The BGP sessions may report in the following states:

State	Listen for TCP	Initiate TCP	TCP Up	Open Sent	Open Received	Neighbor Up
Idle	No					
Connect	Yes					
Active	Yes	Yes				
Open sent	Yes	Yes	Yes	Yes		

Table Continued

Open confirm	Yes	Yes	Yes	Yes	Yes	
Established	Yes	Yes	Yes	Yes	Yes	Yes

Injecting routes/prefixes into the BGP table

The BGP Routing Information Base (RIB) holds the network layer reachability information (NLRI) learned by BGP, as well as the associated path attributes (PAs). An NLRI is simply an IP prefix and prefix length.

A BGP router adds entries to its local BGP table by using the `network` command. The router receives the prefixes through an Update message from a neighbor or by redistributing from another routing protocol.

The `network` command instructs BGP protocol to perform the following steps:

- Look for a route in the current IP routing table that matches the parameters of the `network` command. If the IP route exists, put the equivalent NLRI into the local BGP table.
- With this logic, connected routes, static routes, or IGP routes can be taken from the IP routing table and placed into the BGP table for later advertisement.

The BGP `redistribute` command can redistribute static, connected, and IGP learned routes.

Path attributes

BGP attaches path attributes (PAs) associated with each network path. The PAs provide BGP with granularity and control of routing policies within BGP. The BGP prefix PAs are classified as follows:

- Well-known mandatory
- Well-known discretionary
- Optional transitive
- Optional non-transitive

Per RFC 4271, well-known attributes must be recognized by all BGP implementations. Well-known mandatory attributes must be included with every prefix advertisement, while well-known discretionary attributes may or may not be included.

Optional attributes do not have to be recognized by all BGP implementations. Optional attributes can be set to be transitive and stay with the route advertisement from AS to AS. Other PAs are nontransitive and cannot be shared from AS to AS. BGP PAs are summarized in the following table.

Path Attribute	Description	Characteristics
AS_PATH	Lists ASNs through which the route has been advertised.	Well-known mandatory
NEXT_HOP	Lists the next-hop IP address used to reach an NLRI.	Well-known mandatory

Table Continued

LOCAL_PREF	Used to communicate a BGP router degree of preference for an advertised route. Used only in updates between internal BGP peers.	Well-known discretionary
MULTI_EXIT_DISC	Known as the MED, used to influence incoming traffic only in EBGp updates. Allows an AS to inform another AS of its preferred ingress point.	Optional non-transitive
AGGREGATOR	Lists the RouterId and ASN of the router that created the summary NLRI and indicates where the Aggregation occurred.	Optional transitive
ATOMIC_AGGREGATOR	Tags a summary NLRI as being a summary and indicates that a loss of path information has occurred.	Well-known discretionary
ORIGIN	Describes where the route was imported into BGP; i (IGP), e (EGP), or ? (incomplete information).	Well-known mandatory
COMMUNITY	Identifies a destination as a member of some community of destinations that share one or more common properties. Used to simplify policy enforcement.	Optional transitive
ORIGINATOR_ID	Used by route reflectors (RRs) to denote the router-id of the iBGP neighbor that injected the NLRI into the AS	Optional non-transitive
CLUSTER_LIST	Used by route reflectors (RRs) to list a sequence of route reflection cluster IDs through which the route has passed in order to prevent loops.	Optional non-transitive
MP_REACH_NLRI	Carries the set of reachable destinations together with the next hop information to be used for forwarding to these destinations.	Optional non-transitive
MP_UNREACH_NLRI	Used to withdraw unreachable routes.	Optional non-transitive

BGP best-path calculation

When a BGP router learns multiple routes to the same NLRI, it must choose a single best route. The following list defines the BGP route selection decision process.

1. **Highest administrative weight:** Administrative weight can be assigned to each NLRI locally on a router and the value cannot be communicated to another router. A higher value indicates a better route.
2. **Highest LOCAL_PREF PA:** This attribute can be set on a router inside an AS and distributed inside the AS. A higher value indicates a better route.
3. **Locally injected routed:** Pick the route injected into BGP locally using the network command `redistribution`.
4. **Shortest AS_PATH length:** A shorter AS_PATH indicates a better route. This attribute counts each ASN in the AS_SEQUENCE as one.
5. **Origin PA:** IGP (I) routes are preferred over EGP (E) routes, which are in turn preferred over incomplete (?) routes.
6. **Smallest Multi-Exit Discriminator (MED) PA:** A smaller value indicates a better route.
7. **Neighbor Type:** External BGP (eBGP) routes are better than internal BGP (iBGP) routes.
8. **IGP metric for reaching the NEXT_HOP:** When comparing IGP metrics for each NLRI's NEXT_HOP, a smaller value indicates a better route.
9. **Keep the oldest eBGP route:** If the routes being compared are eBGP and one is the best path, retain the existing best path. This action reduces eBGP route flaps.
10. **Choose the smallest neighbor router ID:** Use the route with the smallest next-hop router router-id. Only perform this step if `bgp bestpath compare-routerid` is configured.
11. **Smallest neighbor ID:** The local router has at least two neighbor relationships with a single other router. For this case, the router prefers the route advertised by the lowest neighbor ID.

Loop prevention

BGP is a path vector routing protocol and does not contain a complete topology of the link state routing protocols.

The BGP attribute AS_PATH is a mandatory attribute and includes a complete listing of all ASNs that the prefix advertisement has traversed from its source AS. AS_PATH is used as a loop prevention mechanism in the BGP protocol. If a BGP router receives a prefix advertisement with its AS listed in the AS_PATH, it discards the prefix because the router thinks the advertisement forms a loop.

Route policies

Route policy is a method that defines how routes are accepted on a router and how routes are advertised. Filtering of routes within BGP is accomplished with route maps, prefix lists, aspath lists, or community lists.

Resetting BGP sessions

When a BGP router configuration is changed, it is often necessary to reset the connections to the affected neighbors in order for the change to take effect. The user must reset the BGP connection for all:

- Changes to BGP related route maps, prefix lists, aspath lists, and community-lists
- Changes to the BGP-related timer specification
- Changes to BGP-related weights

BGP supports two methods of clearing a BGP session:

- A hard reset, which tears down the BGP session, removes BGP routes from the peer, and is the most disruptive.
- A soft reset, which invalidates the BGP cache and requests a full advertisement from its BGP peer.

Initiate a hard or soft reset with the `clear bgp` commands. Clear sessions with all BGP neighbors by using an asterisk `*` instead of the peer IP address.

When a BGP route policy changes, the BGP table must be processed again to notify neighbors. If the BGP session supports route refresh capability, the peer refreshes the prefixes to the requesting router, allowing the inbound policy to process using the new policy changes. The route refresh capability is negotiated when the session is established.

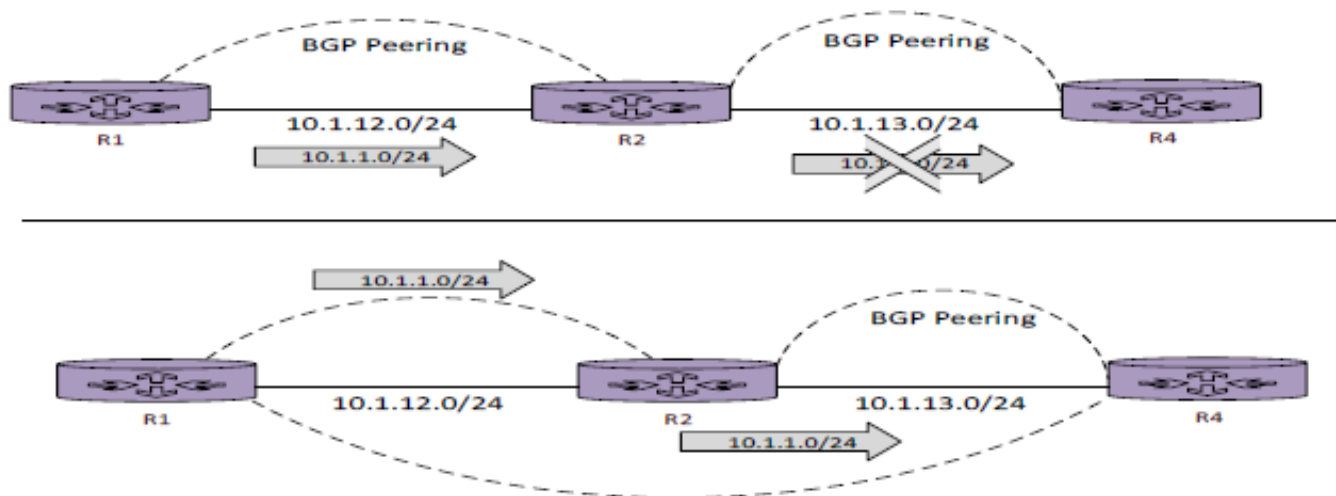
If the BGP session does not support route refresh, use inbound soft reconfiguration so that updates to inbound route policies can be applied without performing a hard reset.

- Inbound soft reconfiguration does not purge the Adj- RIB-In table after routes process into the Loc-RIB table.
- The Adj-RIB-In table maintains the raw unedited routes received from neighbors and allows the inbound route policies to be processed again.
- Enabling this feature can consume a significant amount of memory because the Adj-RIB-In table stays in memory.

IBGP full mesh requirement

BGP uses the AS_PATH as a loop detection and prevention mechanism because the ASN is prepended when advertising to an EBGp neighbor. IBGP peers do not prepend their ASN to the AS_PATH because the NLRIs would fail the validity check and would not install the prefix in the IP routing table.

No other method exists to detect loops with IBGP sessions. RFC 4271 prohibits the advertisement of an NLRI received from one IBGP peer to another. RFC 4271 states that all BGP routers within a single AS must be fully meshed to provide a complete loop-free routing table and prevent traffic black holing.



In the figure above, R1, R2, and R3 are all within AS65100.

- R1 has an IBGP session with R2 and R2 has an IBGP session with R3.
- R1 advertises the 10.1.1.0/24 prefix to R2, which is processed and inserted into the R2 BGP table.
- R2 does not advertise the 10.1.1.0/24 NLRI to R3 because it received the prefix from an IBGP peer.
- To resolve this issue, R1 must form a multi-hop IBGP session so that R3 can receive the 10.1.1.0/24 prefix directly from R1.

Route reflectors

The inability of BGP to advertise a prefix learned from one IBGP peer to another can lead to scalability issues within an AS. The formula $n(n-1)/2$ provides the number of sessions required where n represents the number of routers. A full mesh topology of 10 routers requires 45 sessions. IBGP scalability becomes an issue for large networks.

RFC 1966 specifies that IBGP peering can be configured so that it reflects routes to another IBGP peer. The router reflecting route is known as a route reflector (RR) and the router receiving reflected routes is a route reflector client. Three basic rules apply to route reflectors and route reflection

1. If an RR receives an NLRI from a non-RR client, the RR advertises the NLRI to an RR client. It does not advertise the NLRI to a nonroute reflector client.
2. If an RR receives an NLRI from an RR client, it advertises the NLRI to RR and non-RR clients. Even the RR client that sent the advertisement receives a copy of the route, but it discards the NLRI.
3. If an RR receives a route from an EBGW peer, it advertises the route to RR and non-RR clients.

Loop prevention in route reflectors

Removing the full mesh requirement in an IBGP topology introduces the potential for routing loops. RFC 1966 introduces two other BGP route reflector-specific attributes to prevent loops.

ORIGINATOR_ID is an optional non-transitive BGP attribute created by the first route reflector. This attribute sets the value to the router-id of the router that injected/advertised the route into the AS. If a router receives an NLRI with its router-id in the Originator attribute, the NLRI is discarded.

CLUSTER_LIST is a non-transitive BGP attribute updated by the route reflector. This attribute is appended by the route reflector with its cluster-id. By default this ID is the BGP router-id. The cluster-id can be set with

BGP configuration command `bgp cluster-id <cluster-id>`. If a router receives an NLRI with its cluster-id in the cluster list attribute, the NLRI is discarded.

BGP peer groups

Peer groups can be used to assign common policies and attributes such as an AS number or source-interface for multiple neighbors. A peer group is relevant only to the router on which it is active and is not communicated to router peers. A BGP neighbor can belong to only one peer group. Peer-group activate is not supported in the current release. Peers under the peer groups must be activated individually under the address families.

BGP communities

BGP communities provide additional capability for tagging routes and modifying BGP routing policy on upstream and downstream routers. They can be appended, removed, or modified on each attribute as the route travels from router to router.

BGP communities are an optional transitive BGP attribute that can traverse from autonomous system to autonomous system. A BGP community is a 32-bit number that can be included with a route. It can be displayed as two 16-bit numbers (0-65535):(0-65535).

Aggregate routes

Aggregating routes conserves router resources and accelerates best path calculation by reducing the size of the table. The two methods for BGP route aggregation are:

- **Static:** Create a static reject route for the prefix and advertise the network through a `network` command. The disadvantage of this method is that the summary route will always be advertised even if the networks are not available.
- **Dynamic:** Configure an aggregation network range using the `aggregate-address` command. When viable routes that match the network range enter the BGP table, an aggregate route is created. On the originating router, a reject route for the aggregated prefix is automatically created by BGP as a loop-prevention mechanism.

BGP Graceful-Restart and high availability

The BGP Graceful-Restart (GR) feature allows a BGP speaker to express its ability to preserve forwarding state during:

- BGP protocol (or daemon) restart
- Management Module (MM) switchover

BGP GR is enabled by default and announces GR capability in the BGP OPEN message to peers. BGP initiates the graceful-restart process when an MM switchover occurs and also acts as a GR-aware device.

A GR-aware device, also known as GR helper mode, is notified that the peer router is transitioning and takes appropriate actions based on configuration or default timers.

When a BGP restart happens on the peer router or when MM switchover occurs, the routes currently held in the forwarding table are marked as stale. Thus the forwarding state is preserved as the control plane and the forwarding plane operates independently.

On the restarting peer on which the switchover occurred, BGP on the newly active MM starts to establish sessions with all configured peers. BGP on the nonrestarting side sees new connection requests coming in

while BGP is in an established state. Such an event is an indication for the nonrestarting peer that the peer has restarted. At this point, the restarting peer sends the GR capability with Restart bit set to 1 and Forwarding State bit set to 1.

The nonrestarting peer:

- Cleans up old BGP sessions and marks all the routes in the BGP table that are received from the restarting peer as stale
- Purges all stale routes after the Restart Time expires (if the restarting peer never re-establishes the BGP session)
- Sends an initial routing table update, followed by an End-of-RIB (EoR)

The restarting peer:

- Delays best-path calculation until after receiving EoR from all peers
- Generates updates for its peers and sends the EoR marker after the initial table is sent

The nonrestarting peers:

- Receive the routing updates from the restarting peer
- Remove stale marking for any refreshed route
- Purge any remaining stale routes after EoR is received from the restarting peer or the stale path timer expires

Basic BGP configuration

A BGP session between routers is configured as follows:

1. Create the BGP routing instance and specify the local AS number with the global command `router bgp as-number`.
2. Specify a neighbor and the neighbor AS number with the `neighbor ip-address remote-as as-number` command.
3. Activate the neighbor under the address-family to exchange address-family specific information (capability advertisement, NLRI's etc.).

Address families

MP-BGP adds support for Address families and Sub Address families in addition to IPv4 Unicast which was supported by base BGP (RFC 4271). The 10.02 release supports IPv4 Unicast and IPv6 Unicast address families. Use the `address-family {<ipv4> | <ipv6>} unicast` commands to enter the address-family configuration mode and execute address-family specific configurations.

Scale limits

- Maximum neighbors supported across all VRFs: 256
- Maximum equal cost paths supported: 8
- Maximum route reflector clients allowed across all VRFs: 16

- Maximum BGP routes supported: 256,000
- Maximum routes accepted from a BGP peer: 256,000
- Maximum peer groups allowed across all VRFs: 50
- Maximum AS numbers in as-path attribute: 32
- Maximum as-path entries in a single aspath-list: 128
- Maximum aspath-lists: 128
- Maximum community entries in a single community-list: 128
- Maximum community-lists: 256
- Maximum prefix entries in a single prefix-list: 128
- Maximum prefix-lists: 256
- Maximum route map entries in a single route-map: 128
- Maximum route-maps: 256

BGP commands

address-family

Syntax

```
address-family {ipv4 | ipv6} unicast
```

```
no address-family {ipv4 | ipv6} unicast
```

Description

Specifies address family to use and changes to the configuration context for the specified family:

- `config-bgp-ipv4` for IPv4
- `config-bgp-ipv6` for IPv6
- `config-bgp-ipv4-uc` for IPv4 unicast
- `config-bgp-ipv6-uc` for IPv6 unicast

The `no` form of this command removes the specified address family configuration.

Command context

```
config-bgp
```

Parameters

ipv4

Selects the IPv4 address family.

ipv6

Selects the IPv6 address family.

unicat

Specifies unicast addresses.

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the address family to IPv4 unicast .

```
switch(config-bgp) # address-family ipv4 unicast  
switch(config-bgp-ipv4-uc) #
```

aggregate-address

Syntax

```
aggregate-address <IP-ADDR>/<MASK> [as-set] [summary-only]  
                  [suppress-map <MAP-NAME>] [advertise-map <MAP-NAME>]  
                  [attribute-map <MAP-NAME>]  
  
no aggregate-address <IP-ADDR>/<MASK> [as-set] [summary-only]  
                  [suppress-map <MAP-NAME>] [advertise-map <MAP-NAME>]  
                  [attribute-map <MAP-NAME>]
```

Description

Creates an aggregate address entry in the BGP routing table.

The **no** form of this command removes the specified aggregate address entry.

Command context

```
config-bgp-ipv4-uc  
config-bgp-ipv6-uc
```

Parameters

<ADDRESS>

Specifies an aggregate address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255, or IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (x), where x is a decimal number from 0 to 128.

as-set

The AS_PATH attribute advertised for this route will contain an AS_SET consisting of all AS numbers contained in all paths that are being summarized.

summary-only

Creates the aggregate route but also suppresses advertisements of more-specific routes to all neighbors.

suppress-map <MAP-NAME>

Specifies an aggregate route for creation, but suppresses the advertisement of the created route. Match clauses of route maps can be used to suppress some more-specific routes of the aggregate selectively, and leave others unsuppressed. IP prefix lists and as_path lists match clauses are supported.

advertise-map <MAP-NAME>

Specifies routes that will be used to build attributes of the aggregate route, such as AS_SET or community.

attribute-map <MAP-NAME>

Specifies that the attributes of the aggregate route can be changed.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # aggregate-address 10.0.0.0/8
switch(config-bgp-ipv4-uc) # no aggregate-address 10.0.0.0/8
```

```
switch(config-bgp-ipv6-uc) # aggregate-address 2001:0db8:85a3::8a2e:0370:7334/24
switch(config-bgp-ipv6-uc) # no aggregate-address 2001:0db8:85a3::8a2e:0370:7334/24
```

```
switch(config-bgp-ipv4-uc) # aggregate-address 10.0.0.0/8 as-set summary-only
switch(config-bgp-ipv4-uc) # aggregate-address 10.0.0.0/8 attribute-map RMap
```

```
switch(config-bgp-ipv6-uc) # aggregate-address 2001:0db8:85a3::8a2e:0370:7334/24 as-set summary-only
switch(config-bgp-ipv6-uc) # aggregate-address 2001:0db8:85a3::8a2e:0370:7334/24 attribute-map RMap
```

bgp always-compare-med

Syntax

```
bgp always-compare-med
```

```
no bgp always-compare-med
```

Description

Enables comparison of the Multi Exit Discriminator (MED) for paths from neighbors in different autonomous systems. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The **no** form of this command sets comparison of MED to the default setting (disabled).

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- MED is one of the parameters that is considered when selecting the best path among many alternative paths. The path with a lower MED is preferred over a path with a higher MED.
- During the best-path selection process, MED comparison is done only among paths from the same autonomous system. Use the command **bgp always-compare-med** to change this behavior by

enforcing MED comparison between all paths, regardless of the autonomous system from which the paths are received.

Examples

```
switch(config-bgp)# bgp always-compare-med  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp always-compare-med  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp asnotation dotted

Syntax

```
bgp asnotation dotted
```

```
no bgp asnotation dotted
```

Description

Specifies that Autonomous System (AS) numbers greater than 65535 be shown in dotted integer format for all show commands, including running-configuration.

The `no` form of this command restores the default format of non-dotted, simple integer.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch(config-bgp)# bgp asnotation dotted
```

bgp asnotation dotted-plus

Syntax

```
bgp asnotation dotted-plus
```

```
no bgp asnotation dotted-plus
```

Description

Specifies that all Autonomous System (AS) numbers be shown in dotted integer format for all show commands, including running-configuration.

The `no` form of this command restores the default format of non-dotted, simple integer.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch(config-bgp) # bgp asnotation dotted-plus
```

bgp bestpath as-path ignore

Syntax

```
bgp bestpath as-path ignore
```

```
no bgp bestpath as-path ignore
```

Description

Configures BGP to avoid considering the autonomous system (AS) path during best path route selection. By default, the AS-path is considered during BGP best path selection.

Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command restores default behavior which configures BGP to consider the AS-path during route selection.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # bgp bestpath as-path ignore  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp) # no bgp bestpath as-path ignore  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp bestpath as-path multipath-relax

Syntax

```
bgp bestpath as-path multipath-relax
```

```
no bgp bestpath as-path multipath-relax
```

Description

Configures Border Gateway Protocol (BGP) to treat two BGP routes as equal cost even if their AS-paths differ, as long as their AS-path lengths and other relevant attributes are the same. This allows routes with different AS-paths to be programmed into the forwarding table as equal cost multipath routes.

Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command restores the default behavior which configures BGP to treat two BGP routes as different costs when their AS-paths differ.

Command context

config-bgp

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp bestpath as-path multipath-relax  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp bestpath as-path multipath-relax  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp bestpath compare-routerid

Syntax

```
bgp bestpath compare-routerid
```

```
no bgp bestpath compare-routerid
```

Description

Configures a BGP routing process to compare identical routes received from different external peers during the best path selection process and selects the route with the lowest router ID as the best path. Defaults to disabled.

Any changes in BGP configuration are applied by restarting the current BGP sessions in the VRFs.

The `no` form of this command returns the BGP routing process to the default operation. By default, BGP selects the route that was received first when two routes with identical attributes are received.

Command context

config-bgp

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp bestpath compare-routerid  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp bestpath compare-routerid  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp bestpath med confed

Syntax

```
bgp bestpath med confed
```

```
no bgp bestpath med confed
```

Description

Compares the identical routes received from the different confederation peers and selects the route with the lowest Multi Exit Discriminator (MED) value as the best path. This behavior is disabled by default.

The `no` form of this command prevents the routing process from considering the MED value.



NOTE: The selection of other attributes like `as-multi-path relax` and `as-path ignore` will not affect the behavior of this command within a confederation.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Selecting the route with lowest MED value:

```
switch(config-bgp)# bgp bestpath med confed  
All active BGP sessions in the VRF %s will be restarted.  
Do you want to continue (y/n)?
```

Preventing the routing process from selecting the MED value:

```
switch(config-bgp)# no bgp bestpath med confed  
All active BGP sessions in the VRF %s will be restarted.  
Do you want to continue (y/n)?
```

bgp bestpath med missing-as-worst

Syntax

```
bgp bestpath med missing-as-worst
```

```
no bgp bestpath med missing-as-worst
```

Description

Configures a BGP routing process to assign a value of infinity (max possible) to routes that are missing the Multi Exit Discriminator (MED) attribute. The path without a MED value is the least desirable path. Any changes in BGP configuration are applied by restarting the current BGP sessions in the VRFs.

The `no` form of this command restores default behavior. The default behavior assigns a value of 0 to the missing MED.

Command context

```
config-bgp
```


Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp bestpath med missing-as-worst  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp bestpath med missing-as-worst  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp cluster id

Syntax

```
bgp cluster-id {<IPv4-ADDR> | <ID>}
```

```
no bgp cluster-id {<IPv4-ADDR> | <ID>}
```

Description

Specifies the cluster ID when the BGP router is used as a route-reflector. The cluster ID default is the router ID. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The **no** form of this command sets the cluster ID to the default value, which is the router ID.

Command context

```
config-bgp
```

Parameters

<IPv4-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255. You can remove leading zeros. For example, the address 192.169.005.100 becomes 192.168.5.100.

<ID>

Specifies the cluster ID as 32-bit number. Range: 1 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp cluster-id 2.2.2.2  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp cluster-id 2.2.2.2  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp confederation

Syntax

```
bgp confederation <AS-NUMBER>
```

```
no bgp confederation<AS-NUMBER>
```

Description

Configures a BGP confederation with the confederation identifier. The group of Autonomous Systems (ASs) will be presented as a single autonomous system with the confederation identifier as the AS number.

The `no` form of the command deletes the BGP confederation identifier.

Command context

```
config-bgp
```

Parameters

<AS-NUMBER>

Sets the identifier for the confederation. Range:1-4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring the BGP confederation with the AS number:

```
switch(config-bgp)# bgp confederation 100
```

Deleting BGP confederation identifier:

```
switch(config-bgp)# no bgp confederation 100  
This will delete BGP confederation identifier on this device.  
Do you want to continue (y/n)?
```

bgp confederation peers

Syntax

```
bgp confederation peers <AS NUMBER>
```

```
no bgp bgp confederation peers <AS NUMBER>
```

Description

Configures BGP confederation peers with both same and different sub-autonomous system to establish an eBGP membership. You can configure a list of AS numbers separated by spaces.

The `no` form of this command disables the peer session and deletes the peer information.

Command context

```
config-bgp
```

Parameters

<AS NUMBER>

Specifies the autonomous system numbers to establish an eBGP membership. Range: 64512-65535.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring peers with ASNs:

```
switch(config-bgp) # bgp confederation peers 64512 64513
```

Disabling peers and deleting the peer information:

```
switch(config-bgp) # no bgp confederation peers 64512
```

bgp dampening

Syntax

```
bgp dampening {{half-life <HALF-TIME> reuse <LOW-THRESHOLD> suppress <HI-THRESHOLD> max-suppress-time <MAX-TIME>}  
| route-map <NAME>}
```

Description

Enables route flap dampening which reduces the propagation of unstable routes in the network.

Command context

Parameters

These parameters can be configured at the router level for specific address families or the same parameters can be configured under a route map which can be applied to dampening command.

half-life <HALF-TIME>

Specifies the half-life time in minutes. When the time expires, the penalty on a route gets reduced exponentially to half its current value. Default: 15.

reuse <LOW-THRESHOLD>

Specifies the lower threshold of penalty. On a suppressed route, when the penalty on a route falls below this value, the route is unsuppressed. Default: 750.

suppress <HI-THRESHOLD>

Specifies the upper threshold of penalty. When the penalty on a flapping route exceeds this value, the route is suppressed. Default: 2000.

max-suppress-time <MAX-TIME>

Specifies the maximum time to keep a route suppressed in minutes. Once this timer expires, the route is unsuppressed. Default: 60.

route-map <NAME>

Specifies the name of a route map.

Authority

Administrators or local user group members with execution rights for this command.

Usage

The dampening algorithm assigns a penalty of 1000 to a flapping route every time the route gets withdrawn. The penalty values accumulate on the route every time it flaps. However, the penalty decays and is reduced to half its value by the half-life time.



NOTE: This feature is not applicable on IBGP routes.

Example

```
switch(config)# router bgp 1
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# bgp dampening

switch(config-bgp-ipv4-uc)# bgp dampening route-map abc

switch(config-bgp-ipv4-uc)# bgp dampening route-map xyz

switch(config-bgp-ipv4-uc)# bgp dampening half-life 10 reuse 100 suppress 250 max-suppress-time 45
```

bgp default local-preference

Syntax

```
bgp default local-preference <NUMBER>
```

```
no bgp default local-preference
```

Description

Default local preference value for BGP learned routes. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command sets the local preference to the default value of 100.

Command context

```
config-bgp
```

Parameters

<NUMBER>

Specifies the local preference value. Range: 0 to 4294967295. Default: 100.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp default local-preference 20
All current BGP sessions in VRF default will be restarted.
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp default local-preference
All current BGP sessions in VRF default will be restarted.
Do you want to continue (y/n)?
```

bgp deterministic-med

Syntax

```
bgp deterministic-med  
  
no bgp deterministic-med
```

Description

Enables comparison of the Multi-Exit Discriminator (MED) attribute when selecting routes advertised by different peers in the same autonomous system. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command sets MED comparison to the default setting of disabled.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp deterministic-med  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp deterministic-med  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp fast-external-fallover

Syntax

```
bgp fast-external-fallover  
  
no bgp fast-external-fallover
```

Description

Sets the switch to reset the BGP sessions of any directly adjacent external peers if the link used to reach them goes down.

The `no` form of this command disables session reset.

Command context

```
config-bgp
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp fast-external-fallover  
switch(config-bgp)# no bgp fast-external-fallover
```

bgp graceful-restart restart-time

Syntax

```
bgp graceful-restart restart-time <DELAY>
```

```
no bgp graceful-restart restart-time
```

Description

Sets the graceful restart timer which determines how long the switch waits for a graceful-restart capable neighbor to re-establish BGP peering. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command resets to the default value of 120 seconds.

Command context

```
config-bgp
```

Parameters

<DELAY>

Graceful restart timer delay in seconds. Range: 1 to 3600. Default: 1500.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Graceful restart functionality is enabled by default, and there is no command to disable the functionality at the protocol level.
- However, the graceful-restart functionality can be disabled globally using the command `router graceful-restart`.

Examples

```
switch(config-bgp)# bgp graceful-restart restart-time 150  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp graceful-restart restart-time  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp graceful-restart stalepath-time

Syntax

```
bgp graceful-restart stalepath-time <TIME>
```

```
no bgp graceful-restart stalepath-time
```

Description

Sets the stale path timer. This timer determines how long BGP keeps stale routes from the restarting BGP peer. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command resets the stale path timer to the default of 300 seconds.

Command context

`config-bgp`

Parameters

<TIME>

Specifies the stale path timer in seconds. Range: 1 to 3600. Default: 300.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp graceful-restart stalepath-time 300  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp graceful-restart stalepath-time  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

bgp log-neighbor-changes

Syntax

`bgp log-neighbor-changes`

`no bgp log-neighbor-changes`

Description

Enables logging of BGP neighbor session state changes.

The `no` form of this command disables logging of changes in BGP neighbor adjacencies.

Command context

`config-bgp`

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# bgp log-neighbor-changes  
switch(config-bgp)# no bgp log-neighbor-changes
```

bgp maxas-limit

Syntax

```
bgp maxas-limit <LENGTH>
```

```
no bgp maxas-limit
```

Description

Specifies the maximum size of AS paths in update messages. Routes with AS paths greater than the specified length are discarded.

The `no` form of this command sets the limit to the default of 32.

Command context

```
config-bgp
```

Parameters

<LENGTH>

Specifies the number of AS segments. Length: 1 to 32 characters. Default: 32.

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch(config-bgp)# bgp maxas-limit 20  
switch(config-bgp)# no bgp maxas-limit
```

bgp router-id

Syntax

```
bgp router-id <ROUTER-ID>
```

```
no bgp router-id <ROUTER-ID>
```

Description

Configures a fixed router ID for the BGP peer process running on the router. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command removes the fixed router ID from the running configuration and restores the default router ID selection.

Command context

```
config-bgp
```

Parameters

<ROUTER-ID>

Specifies the router ID in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

If router-id is changed, then all the active BGP peer sessions go down and restart with the newly configured router-id.

Authority

Administrators or local user group members with execution rights for this command.

Usage

BGP determines the router ID as follows:

1. The address configured with the command `bgp router-id`.
2. The highest IP address on all the loopback interfaces.
3. The highest IP address on any interface.

Examples

```
switch(config-bgp)# bgp router-id 1.1.1.1  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

```
switch(config-bgp)# no bgp router-id 1.1.1.1  
All current BGP sessions in VRF default will be restarted.  
Do you want to continue (y/n)?
```

clear bgp

Syntax

```
clear bgp [vrf <VRF-NAME>][ipv4 unicast | ipv6 unicast | l2vpn evpn | all]  
        { * | <NEIGHBOR-IP-ADDR> } [soft in]
```

Description

Resets BGP peer sessions. Sends a route refresh request when you have specified `soft in`. Optionally, you can specify reset for a specific VRF.

Command context

Operator (>) or Manager (#)

Parameters

ipv4 unicast

Specifies the IPv4 address family.

ipv6 unicast

Specifies the IPv6 address family.

l2vpn evpn

Selects the L2VPN EVPN address family

vrf <VRF-NAME>

Specifies a VRF name.

all

Specifies all VRFs and address families.

*** | <NEIGHBOR-IP-ADDRESS>**

Specifies a neighbor IP address for which peer sessions are to be reset, or * to reset all sessions.

soft in

Send a route refresh request.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# clear bgp all *
switch# clear bgp ipv4 unicast 192.168.12.1 soft in
```

```
switch# clear bgp l2vpn evpn * soft in
switch# clear bgp l2vpn evpn 9.0.0.2 soft in
```

disable | enable

Syntax

disable | enable

Description

This command disables or enables the BGP instance while retaining the configuration. Disable and enable of the BGP instance may result in a change of the router ID.

By default the BGP instance is enabled.

Command context

config-bgp

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# router bgp 100
switch(config-bgp)# disable
```

distance bgp

Syntax

distance bgp <EXTERNAL> <INTERNAL>

no distance bgp <EXTERNAL> <INTERNAL>

Description

Configures the administrative distance for BGP.

The no form of this command restores the default settings, 20 for eBGP and 200 for iBGP,

Command context

config-bgp-ipv4-uc

config-bgp-ipv6-uc

Parameters

<EXTERNAL>

Specifies the administrative distance for eBGP routes. Range: 1 to 255. Default: 20.

<INTERNAL>

Specifies the administrative distance for iBGP routes. Range: 1 to 255. Default: 200.

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch(config-bgp-ipv4-uc) # distance bgp 100 150
switch(config-bgp-ipv4-uc) # no distance bgp 100 150
```

maximum-paths

Syntax

maximum-paths <MAXPATHS>

no maximum-paths <MAXPATHS>

Description

Configures the maximum number of paths that BGP adds to the route table for equal-cost multipath (ECMP) load balancing for routes learned from both internal and external BGP. Any changes in BGP configuration are applied by restarting the current BGP sessions on the VRFs.

The `no` form of this command restores the default setting of 4.

Command context

config-bgp

Parameters

<MAXPATHS>

Specifies the maximum number of paths. Range: 1 to 8. Default: 4.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config) # router bgp 1
switch(config-bgp) # maximum-paths 4
All current BGP sessions in VRF default will be restarted.
Do you want to continue (y/n)? y
```

```
switch(config-bgp) # no maximum-paths
All current BGP sessions in VRF default will be restarted.
Do you want to continue (y/n)? y
```

neighbor activate

Syntax

```
neighbor <IP-ADDR> activate  
  
no neighbor <IP-ADDR> activate
```

Description

This command enables the address-family capability and exchange of information specific to an address family with a BGP neighbor.

The `no` form of this command removes the address-family capability and disables the exchange of routes for the specified address-family with the BGP neighbor.

Command context

```
config-bgp-ipv4-uc  
config-bgp-ipv6-uc  
config-bgp-l2vpn-evpn
```

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 activate  
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 activate  
  
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 activate  
switch(config-bgp-l2vpn-evpn) # no neighbor 1.1.1.1 activate
```

neighbor advertisement-interval

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} advertisement-interval <INTERVAL>  
  
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} advertisement-interval
```

Description

Sets the advertisement interval, which defines the length of time between transmission of BGP routing updates.

The `no` form of this command restores the default value. Default values are 30 seconds for external BGP peer and 5 seconds for internal BGP peer.

Command context

config-bgp-ipv4-uc

config-bgp-ipv6-uc

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255, or IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<PEER-GROUP-NAME>

Specifies a Peer-Group.

<INTERVAL>

Specifies the advertisement interval in seconds. Range: 0 to 600. Default: 30 for external BGP peer and 5 for internal BGP peer.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 advertisement-interval 20
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 advertisement-interval
```

```
switch(config-bgp-ipv4-uc) # neighbor pg advertisement-interval 50
switch(config-bgp-ipv4-uc) # no neighbor pg advertisement-interval
```

neighbor add-paths

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} add-paths {send | recv | both}
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} add-paths {send | recv | both}
```

Description

Enables additional path capability for BGP as described in RFC 7911. This allows BGP peer to send, receive, or send and receive multiple paths for the same address prefix without the subsequent advertisements implicitly replacing any previous paths. The additional path includes the first (N-1) best paths, which means that the total paths for an address prefix received by a BGP speaker will include the best path and the additional paths determined by its BGP peer.

With additional path feature, each path is identified by a path identifier in addition to the address prefix. To use this command, the backup path of BGP next-hop must be different than the primary path.

The **no** form of this command disables the additional path feature.

Command context

config-bgp-ipv4-uc

or

config-bgp-ipv6-uc

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

add-paths {send | recv | both}

Configures the additional paths in one of the following ways:

- **send**—Enables the neighbor to send the additional paths.
- **recv**—Enables the neighbor to receive the additional paths.
- **both**—Enables the neighbor to send and receive the additional paths.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling BGP neighbor to send the additional paths:

```
switch(config)# router bgp 100
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# neighbor 1.1.1.1 add-paths send
```

Disabling BGP neighbor to send the additional paths:

```
switch(config)# router bgp 100
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# no neighbor 1.1.1.1 add-paths send
```

neighbor add-paths advertise-best

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} add-paths advertise-best <2-4>
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} add-paths advertise-best <2-4>
```

Description

Controls the number of best BGP Paths to be advertised by a BGP speaker to a BGP peer. When enabled, it allows BGP speaker to advertise more than one best paths for the same address prefix. The total paths for an address prefix will include the best path and the additional paths.

The **no** form of this command removes the advertise best path configuration.

Command context

config-bgp-ipv4-uc

or

config-bgp-ipv6-uc

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

advertise-best <2-4>

Specifies the number of best BGP paths to be advertised to a BGP Peer. Range: 2 to 4. Default: 2.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the number of best paths to send to the neighbor:

```
switch(config)# router bgp 100
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# neighbor 1.1.1.1 add-paths advertise-best 3
```

Removing the advertise best path configuration:

```
switch(config)# router bgp 100
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# no neighbor 1.1.1.1 add-paths advertise-best 3
```

neighbor allowas-in

Syntax

```
neighbor {<IP-ADDRESS> | <PEER-GROUP-NAME>} allowas-in <LIMIT>
```

```
no neighbor {<IP-ADDRESS> | <PEER-GROUP-NAME>} allowas-in <LIMIT>
```

Description

Specifies the number of times that the AS path of a received route can contain the AS number of the recipient BGP speaker and still be accepted. When this configuration is applied to a peer-group, all the neighbors that are part of the peer-group inherit this setting.

The **no** form of this command restores the default setting, which is to reject as a loop any route where the path contains the speaker AS number.

Command context

config-bgp-ipv4-uc

config-bgp-ipv6-uc

config-bgp-l2vpn-evpn

Parameters

<IP-ADDRESS>

Specifies the neighbor IP address in the IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255, or in the IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.



NOTE: IPv6 MP-BGP peering must not be used for L2VPN EVPN address family, because VXLAN tunnel interface does not support IPv6 addresses.

<PEER-GROUP-NAME>

Specifies a peer group.

<LIMIT>

Specifies the number of times that the AS path of a received route can contain the AS number of the recipient BGP. Range: 1 to 10.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 allows-in 5
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 allows-in
```

```
switch(config-bgp-ipv6-uc) # neighbor 2001:0db8:85a3::8a2e:0370:7334 allows-in 5
switch(config-bgp-ipv6-uc) # no neighbor 2001:0db8:85a3::8a2e:0370:7334 allows-in
```

```
switch(config-bgp-ipv4-uc) # neighbor PG allows-in 5
switch(config-bgp-ipv4-uc) # no neighbor PG allows-in
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 allows-in 5
switch(config-bgp-l2vpn-evpn) # no neighbor 1.1.1.1 allows-in
```

```
switch(config-bgp-l2vpn-evpn) # neighbor PG allows-in 5
switch(config-bgp-l2vpn-evpn) # no neighbor PG allows-in
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 2001:0db8:85a3::8a2e:0370:7334 allows-in 5
switch(config-bgp-l2vpn-evpn) # no neighbor 2001:0db8:85a3::8a2e:0370:7334 allows-in
```

neighbor capability orf prefix-list

Syntax

```
neighbor <IP-ADDRESS> capability orf prefix-list {send | receive | both}

no neighbor <IP-ADDRESS> capability orf prefix-list {send | receive | both}
```

Description

Enables the Outbound Route filtering (ORF) capability with the neighbor in one of the three available modes. The available modes are send, receive, and both. The ORF capability is executed based on prefix list only.

The Outbound Route Filtering (ORF) capability provides a mechanism for a BGP speaker to send a set of Outbound Route Filters (ORFs) that can be used by its BGP peer to filter its outbound routing updates to the speaker. This is a filtering method used to reduce the computation on the router receiving the route.

The **no** form of this command disables the ORF capability.

Command context

config-bgp-ipv4-uc
or

config-bgp-ipv6-uc

Parameters

<IP-ADDRESS>

Specifies an IP address.

capability orf prefix-list {send | receive | both}

Enables ORF prefix list capability with the neighbor in one of the following modes:

- **send** - Enables the ORF prefix list capability in send mode.
- **receive** - Enables the ORF prefix list capability in receive mode.
- **both** - Enables the ORF prefix list capability in both send and receive mode.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling the ORF prefix list capability in both send and receive mode:

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 capability orf prefix-list both
```

Enabling the ORF prefix list capability in send mode:

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 capability orf prefix-list send
```

Disabling the ORF prefix list capability in both send and receive mode: :

```
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 capability orf prefix-list both
```

neighbor default-originate

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} default-originate [route-map <MAP-NAME>]
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} default-originate [route-map <MAP-NAME>]
```

Description

Enables the local router to send the default route 0.0.0.0 to a neighbor. The neighbor can then use this route to reach the router when all other routes are unavailable. Use the `route-map` option to configure the route map to modify the default route attributes.

The `no` form of this command disables this feature.

Command context

config-bgp-ipv4-uc

config-bgp-ipv6-uc

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255, or IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<PEER-GROUP-NAME>

Specifies a peer group.

<MAP-NAME>

Sets the route map to modify the default route attributes.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 default-originate  
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 default-originate
```

```
switch(config-bgp-ipv4-uc) # neighbor PG default-originate  
switch(config-bgp-ipv4-uc) # no neighbor PG default-originate
```

neighbor ebgp-multihop

Syntax

```
neighbor {<IP-ADDR> | <PEER-GROUP-NAME>} ebgp-multihop <HOP-COUNT>
```

```
no neighbor {<IP-ADDRESS> | <PEER-GROUP-NAME>} ebgp-multihop <HOP-COUNT>
```

Description

Enables BGP to establish a session with external peers residing on networks that are not directly connected. By default, BGP can only establish sessions with external BGP peers that are directly connected.

The neighbor connection must be reset using `clear bgp` to allow this configuration to take effect.

The `no` form of this command disables the peer ebgp-multihop feature.

Command context

```
config-bgp
```

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

ebgp-multihop <HOP-COUNT>

Specifies the maximum number of hops to reach the peer.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling BGP to establish connection with external peers residing on networks that are not directly connected:

```
switch(config-bgp) # neighbor 1.1.1.1 ebgp-multihop 5
switch(config-bgp) # no neighbor 1.1.1.1 ebgp-multihop
```

Disabling BGP to establish connection with external peers residing on networks that are not directly connected:

```
switch(config-bgp) # neighbor pg ebgp-multihop 5
switch(config-bgp) # no neighbor pg ebgp-multihop
```

neighbor fall-over

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} fall-over
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} fall-over
```

Description

Enables BGP fast peering session deactivation. When neighbor fall-over is configured, the BGP process monitors the RIB and if the route to peer is not present in the routing table, it immediately deactivates the peer session without waiting for the hold down timer. It is disabled by default.

The `no` form of this command disables this feature.

Command context

config-bgp

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Usage

Neighbor fall-over does not track connected or static routes to peers. However, this is not an issue when IBGP peering is using a loopback interface. To force a fall-over for connected and static routes, use the command `neighbor fall-over bfd`.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 fall-over
switch(config-bgp) # no neighbor 1.1.1.1 fall-over
```

```
switch(config-bgp) # neighbor PG fall-over
switch(config-bgp) # no neighbor PG fall-over
```

neighbor fall-over bfd

Syntax

```
neighbor {<IPV4-ADDR>|<IPV6-ADDR>|<PEER-GROUP-NAME>} fall-over bfd
```

```
no neighbor {<IPV4-ADDR>|<IPV6-ADDR>|<PEER-GROUP-NAME>} fall-over bfd
```

Description

Enables BGP to register with BFD to receive fast peering session deactivation messages from BFD. You can either configure BFD support for BGP per neighbor or peer-group.

The `no` form of this command disables BGP for BFD.



NOTE: Multihop BFD is not supported for BGP.

BFD over BGP using IPv6 is not supported on 8320 and 8325 switch series.

Command context

```
config-bgp
```

Parameters

<IPV4-ADDR>

Specifies the neighbor address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

<IPV6-ADDR>

Specifies the neighbor address in IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F. This parameter applies only to 6300 and 6400 switch series.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling BFD for BGP neighbor with IPv4 address:

```
switch(config-bgp) # neighbor 1.1.1.1 fall-over bfd
```

Enabling BFD for BGP neighbor with IPv6 address (applies only to 6300 and 6400 switch series):

```
switch(config-bgp) # neighbor 1000::1 fall-over bfd
```

Enabling BFD for peer group:

```
switch(config-bgp) # neighbor PG fall-over bfd
```

Disabling BFD for BGP per neighbor IPv4 address:

```
switch(config-bgp) # neighbor 1.1.1.1 fall-over bfd
```

Disabling BFD for BGP per neighbor with IPv6 address (applies only to 6300 and 6400 switch series):

```
switch(config-bgp)# no neighbor 1000::1 fall-over bfd
```

Disabling BFD for peer group:

```
switch(config-bgp)# no neighbor PG fall-over bfd
```

neighbor graceful-shutdown

Syntax

```
neighbor {<IP-ADDR> | <PEER-GROUP-NAME>} graceful-shutdown
    [ local-preference <LOCAL-PREF>
    | <CONFIG-DELAY>
    | <CONFIG-DELAY> local-preference <LOCAL-PREF> ]

no neighbor {<IP-ADDR> | <PEER-GROUP-NAME>} graceful-shutdown
    [ local-preference <LOCAL-PREF>
    | <CONFIG-DELAY>
    | <CONFIG-DELAY> local-preference <LOCAL-PREF> ]
```

Description

Configures the wait time before shutting down the BGP neighbor session, and can also configure the local preference value to be advertised before graceful shutdown.

The **no** form of this command sets the wait time to the default value of 180 seconds and the local-preference value to the default of 0.

Command context

config-bgp

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (**x.x.x.x**), where **x** is a decimal number from 0 to 255, or IPv6 format (**xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx**), where **x** is a hexadecimal number from 0 to F.

<PEER-GROUP-NAME>

Specifies a peer group.

local-preference <LOCAL-PREF>

Specifies the local preference value for exporting the iBGP routes during a graceful shutdown. The lower the value, the lower the local preference. A value of 0 indicates that the route is the least preferred. Range: 0 to 4294967295. Default: 0.

<CONFIG-DELAY>

Specifies the time to wait before shutting down the neighbor in seconds. Range: 10 to 1200. Default: 180.

Authority

Administrators or local user group members with execution rights for this command.

Usage

If the graceful shutdown timer has already started and the administrator configures a command that triggers a session restart, traffic loss can occur if the graceful shutdown delay is not sufficient for the BGP peers to converge to a new route.



NOTE: On each Autonomous System Boundary Router (ASBR) supporting the graceful shutdown receiver procedure, an inbound BGP route policy must be applied on all EBGp sessions of the ASBR.

The policy must match the GSHUT community and lower the precedence of the route by changing the route attributes.

The Graceful-Shutdown feature does not work for reflected routes because the route reflector (RR) does not modify local-preference attribute. The routes, originated by the RR, carry the GSHUT local-preference value. As per the RFC 4456, when an RR reflects a route, it should not modify the following path attributes:

- NEXT-HOP
- AS-PATH
- LOCAL-PREF
- MED

Their modification could potentially result in routing loops. In this situation, apply on the RR an inbound BGP route policy, meeting the following conditions:

- Match the graceful-shutdown community.
- Set the local preference attributes of the paths tagged with the graceful-shutdown community to a lower value than other routes to the same destination.

Examples

Setting the wait time delay:

```
switch(config-bgp) # neighbor 1.1.1.1 graceful-shutdown 10
```

Setting the local-preference value:

```
switch(config-bgp) # neighbor 1.1.1.1 graceful-shutdown local-preference 100
```

Setting the wait time delay and local-preference value:

```
switch(config-bgp) # neighbor 1.1.1.1 graceful-shutdown 10 local-preference 100
```

Setting the wait time delay to the default of 180 seconds:

```
switch(config-bgp) # no neighbor 1.1.1.1 graceful-shutdown 10
```

Setting the local-preference value to default of 0:

```
switch(config-bgp) # no neighbor 1.1.1.1 graceful-shutdown local-preference 100
```

Setting the wait time delay and local-preference value to defaults:

```
switch(config-bgp) # no neighbor 1.1.1.1 graceful-shutdown 10 local-preference 100
```

Complete deletion:

```
switch(config-bgp) # no neighbor 1.1.1.1 graceful-shutdown
```

neighbor local-as

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} local-as <AS-NUMBER>
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} local-as
```

Description

Configures an alternate local AS number that can be used to establish a session with a peer, allowing a router to appear to be a member of a second autonomous system (AS), and its real AS.

Local AS allows two autonomous systems to merge without modifying peering arrangements. This command is valid only for external peers.

The `no` form of this command restores the default, which is for a peering session to be established using the primary AS (primary AS is the AS number specified at the time of neighbor creation using the command `neighbor remote-as`).

Command context

```
config-bgp
```

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

local-as <AS-NUMBER>

Specifies a 4-byte AS number in asplain format (z), or asdot format (x.y), where z is a number from 1 to 4294967295 and x and y are 16-bit numbers.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 local-as 200
switch(config-bgp) # no neighbor 1.1.1.1 local-as
```

```
switch(config-bgp) # neighbor pg local-as 200
switch(config-bgp) # no neighbor pg local-as
```

neighbor maximum-prefix

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>}
    maximum-prefix <MAXIMUM> [threshold <THRESHOLD>]
    [restart <INTERVAL>] [warning-only]
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} maximum-prefix
```

Description

Sets the maximum number of prefixes that can be received from a neighbor.

By default, the device accepts 128,000 prefixes from a BGP neighbor with a threshold value of 75%. A warning message is generated when the number of prefixes per neighbor reaches 75% of default prefix limit. Another warning message is generated when the default prefix limit is reached.

The session is re-established only if the number of routes received from the BGP peer does not exceed the configured prefix limit. When the restart timer is configured, sessions are automatically re-established when the timer expires.

The `no` form of this command disables the maximum number of prefixes limit.

Command context

`config-bgp-ipv4-uc`

`config-bgp-ipv6-uc`

Parameters

<IP-ADDRESS>

Specifies the IP address of the neighbor in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<PEER-GROUP-NAME>

Specifies a Peer-Group.

maximum-prefix <MAXIMUM>

Specifies the maximum number of prefixes allowed from the specified neighbor. Range: 1 to 128000. Default: 128000.

threshold <THRESHOLD>

Specifies at what percentage of `MAXIMUM` a warning message is generated. Range: 1 to 100. Default: 75.

For example, if `MAXIMUM` is set to 1000 and `threshold` is 70, the router generates a warning message when the number of BGP learned routes from the neighbor exceeds 70 percent of 1000 (700) routes.

restart <INTERVAL>

Specifies interval in seconds for restarting the BGP connection when the prefix limit is exceeded. Range: 30 to 65535.

warning-only

Specifies generating and logging a warning message without disconnecting the BGP session when the prefix limit is exceeded.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the prefix limit to 1000 prefixes:

```
switch(config-bgp-ipv4-uc) # neighbor 10.0.0.1 maximum-prefix 1000
```

Enabling logging of a warning message when more than 1000 prefixes are received:

```
switch(config-bgp-ipv4-uc) # neighbor 10.0.0.1 maximum-prefix 1000 warning-only
```


Setting the prefix limit to 1000 prefixes and enabling logging of a warning message when 500 prefixes are received:

```
switch(config-bgp-ipv4-uc) # neighbor 10.0.0.1 maximum-prefix 1000 threshold 50
```

Setting the prefix limit to 1000 prefixes and enabling logging of a warning message when 500 prefixes are received and a second warning when the prefix limit is exceeded without disconnecting the session:

```
switch(config-bgp-ipv4-uc) # neighbor 10.0.0.1 maximum-prefix 1000 threshold 50 warning-only
```

Removing the threshold value:

```
switch(config-bgp-ipv4-uc) # no neighbor 10.0.0.1 maximum-prefix 1000 threshold 50
```

Disabling the maximum-prefix feature:

```
switch(config-bgp-ipv4-uc) # no neighbor 10.0.0.1 maximum-prefix
```

neighbor next-hop-self

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} next-hop-self
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} next-hop-self
```

Description

Configures the router as the next hop for a BGP-speaking neighbor or peer group, and enables BGP to send itself as the next hop for advertised routes.

The **no** form of this command resets the peer nexthop-self status to default. The next hop will be generated based on the IP.

Command context

```
config-bgp-ipv4-uc
```

```
config-bgp-ipv6-uc
```

Parameters

<IP-ADDR>

Specifies the neighbor's IP address in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255, or IPv6 format (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx), where x is a hexadecimal number from 0 to F.

<PEER-GROUP-NAME>

Specifies a peer group.

all

Applies the configuration to all route-reflector clients.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- An administrator uses this command to make a BGP speaker fill its address when advertising routes to a BGP peer.
- This command is useful in non-meshed networks where BGP neighbors may not have direct access to all other neighbors on the same IP subnet.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 next-hop-self
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 next-hop-self

switch(config-bgp-ipv6-uc) # neighbor 2001:0db8:85a3::8a2e:0370:7334 next-hop-self
switch(config-bgp-ipv6-uc) # no neighbor 2001:0db8:85a3::8a2e:0370:7334 next-hop-self

switch(config-bgp-ipv4-uc) # neighbor pg next-hop-self
switch(config-bgp-ipv4-uc) # no neighbor pg next-hop-self
```

neighbor next-hop-unchanged

Syntax

```
neighbor <IP-ADDRESS> next-hop-unchanged

no neighbor <IP-ADDRESS> next-hop-unchanged
```

Description

Enables the neighbor to preserve next-hop while advertising routes to eBGP peers, in the L2VPN EVPN address-family.

The `no` form of this command resets the peer next-hop-unchanged status to default.

Command context

```
config-bgp-l2vpn-evpn
```

Parameters

<IP-ADDRESS>

Specifies the neighbor IP address in the IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or in the IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.



NOTE: IPv6 MP-BGP peering must not be used for L2VPN EVPN address family, because VXLAN tunnel interface does not support IPv6 addresses.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 next-hop-unchanged

switch(config-bgp-l2vpn-evpn) # neighbor 2001:0db8:85a3::8a2e:0370:7334 next-hop-unchanged
```

neighbor orf prefix-list in

Syntax

```
neighbor <IP-ADDRESS> orf-prefix-list <PREFIX-LIST-NAME> in  
  
no neighbor <IP-ADDRESS> orf-prefix-list <PREFIX-LIST-NAME> in
```

Description

Applies an inbound prefix list filter to filter the distribution of BGP neighbor information.
The `no` form of this command restores the default behavior of not applying the prefix list filter.



NOTE: This command must be used only along with the ORF capability to take effect.

Command context

```
config-bgp-ipv4-uc  
or  
config-bgp-ipv6-uc
```

Parameters

<IP-ADDRESS>

Specifies an IP address.

orf-prefix-list <PREFIX-LIST-NAME>

Sends the prefix list name to be filtered.

Usage

To use this command, the following conditions must be met:

- If route-map inbound is also applied on multiple neighbors along with ORF, then the route-map name must be common on all the neighbors.
- If route-map inbound is also applied on an IPv6 AF BGP neighbor, then the route-map sequence number with value 1 cannot be used.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Applying the inbound prefix list filter:

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 orf-prefix-list ABC in
```

Removing the inbound prefix list filter:

```
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 orf-prefix-list ABC in
```

neighbor passive

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} passive
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} passive
```

Description

Configures a specific neighbor, or all neighbors in a peer-group, as passive, which means that they will not initiate the TCP session.

The neighbor connection must be reset using `clear ip bgp` for this setting to take effect.

The `no` form of this command enables the neighbor to initiate the TCP session.

Command context

```
config-bgp
```

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 passive
switch(config-bgp) # no neighbor 1.1.1.1 passive
```

```
switch(config-bgp) # neighbor pg passive
switch(config-bgp) # no neighbor pg passive
```

neighbor password

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>}
    password {ciphertext | plaintext} <PASSWORD>
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} password
```

Description

Enables message digest5 (MD5) authentication on a TCP connection between two BGP neighbors. When the password is applied to a peer-group, all the neighbors that are part of peer-group inherit the configured setting.

The neighbor connection must be reset using `clear ip bgp` to allow this configuration to take effect.

The `no` form of this command disables this feature.

Command context

config-bgp

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a Peer-Group.

{ciphertext | plaintext}

Specifies configuring the password as either a ciphertext key or plaintext key.

<PASSWORD>

Specifies the password.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 password plaintext hpe_routing  
switch(config-bgp) # no neighbor 1.1.1.1 password
```

```
switch(config-bgp) # neighbor pg password plaintext bgp_hpe_routing  
switch(config-bgp) # no neighbor pg password
```

neighbor port

Syntax

neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} port <NUMBER>

no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} port

Description

Sets a custom TCP port on which to communicate with the BGP neighbor.

When this configuration is applied for peer-group, all the neighbors that are part of peer-group will inherit this setting. Though the neighbor inherits the configuration from the peer-group, the neighbor-specific command, if configured, takes precedence.

This setting only takes effect after a hard reset of the session.

The **no** form of this command allows a random TCP port to be selected for the communication with the BGP neighbor.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

port <NUMBER>

Specifies a TCP port number. Range: 0 to 65535.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 port 1500
switch(config-bgp) # no neighbor 1.1.1.1 port
```

```
switch(config-bgp) # neighbor PG port 1500
switch(config-bgp) # no neighbor PG port
```

neighbor remote-as

Syntax

```
neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} remote-as <AS-NUMBER>
```

```
no neighbor {<IP-ADDR>|<PEER-GROUP-NAME>} remote-as <AS-NUMBER>
```

Description

Creates a peer, initiates the connection to the peer, and adds an entry to the BGP neighbor table. Specifies a neighbor with an autonomous system (AS) number that identifies the neighbor as internal to the local autonomous system. Otherwise, the neighbor is considered as external. By default, neighbors that are defined using this command, exchange only unicast address prefixes.

The **no** form of this command disables the peer session and deletes the peer information.

Command context

config-bgp

Parameters

<IP-ADDR>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

remote-as <AS-NUMBER>

Specifies a 4-byte AS number in asplain format (z), or asdot format (x.y), where z is a number from 1 to 4294967295 and x and y are 16-bit numbers in the range 0 to 65535.

Authority

Administrators or local user group members with execution rights for this command.

Usage

The configured peer AS number is compared with the AS number received in the open message and a peer session is initiated only if both the AS numbers match.

Examples

```
switch(config-bgp)# neighbor 1.1.1.1 remote-as 1
switch(config-bgp)# no neighbor 1.1.1.1 remote-as 1
```

```
switch(config-bgp)# neighbor pg remote-as 1
switch(config-bgp)# no neighbor pg remote-as 1
```

neighbor remove-private-AS

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} remove-private-AS
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} remove-private-AS
```

Description

This command forces the BGP neighbor to drop the private AS numbers. When the outbound update contains a sequence of private AS numbers, this sequence is dropped. If the command is configured for peer-group, then all the neighbors that are part of peer-group will remove the private-AS before sending the BGP update message.

The `no` form of this command allows the private-AS number to be carried in BGP update message.

The neighbor connection must be reset using `clear ip bgp neighbor-ip-address` to allow this configuration to take effect.

Command context

```
config-bgp
```

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# neighbor 1.1.1.1 remove-private-AS
switch(config-bgp)# no neighbor 1.1.1.1 remove-private-AS
```

```
switch(config-bgp)# neighbor PG remove-private-AS
switch(config-bgp)# no neighbor PG remove-private-AS
```

neighbor route-map

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} route-map <MAP-NAME> {in|out}
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} route-map <MAP-NAME> {in|out}
```

Description

This command applies a route map to incoming or outgoing routes. It configures the route map for modifying the default attributes of the route.



NOTE: A peer group configuration for outbound route map will override the configuration of the neighbor.

The `no` form of this command removes a route map.

Command context

`config-bgp-ipv4-uc`

or

`config-bgp-ipv6-uc`

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

<MAP-NAME>

Specifies the name of the route map.

in|out

Sets the route map policy to apply to either the received routes from the neighbor (`in`) or the advertised routes to the neighbor (`out`).

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 route-map HPE in
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 route-map HPE in
```

```
switch(config-bgp-ipv4-uc) # neighbor PG route-map HPE in
switch(config-bgp-ipv4-uc) # no neighbor PG route-map HPE in
```

```
switch(config) # route-map Rmap permit seq 10
```

```
switch(config-route-map-Rmap-10) # match metric 100
```

```
switch(config-route-map-bgp-10) # router bgp 100
```

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 remote-as 100
```

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 route-map Rmap out
```


neighbor route-reflector-client

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} route-reflector-client
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} route-reflector-client
```

Description

This command configures the router as a BGP route reflector and the specified peer as its client. The `no` form of this command disables this function.

Command context

```
config-bgp-ipv4-uc
```

or

```
config-bgp-ipv6-uc
```

or

```
config-bgp-l2vpn-evpn
```

Parameters

<IP-ADDRESS>

Specifies the neighbor IP address in the IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or in the IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.



NOTE: IPv6 MP-BGP peering must not be used for L2VPN EVPN address family, because VXLAN tunnel interface does not support IPv6 addresses.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 route-reflector-client
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 route-reflector-client
```

```
switch(config-bgp-ipv4-uc) # neighbor PG route-reflector-client
switch(config-bgp-ipv4-uc) # no neighbor PG route-reflector-client
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 route-reflector-client
switch(config-bgp-l2vpn-evpn) # no neighbor 1.1.1.1 route-reflector-client
```

```
switch(config-bgp-l2vpn-evpn) # neighbor PG route-reflector-client
switch(config-bgp-l2vpn-evpn) # no neighbor PG route-reflector-client
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 2001:0db8:85a3::8a2e:0370:7334 route-reflector-client
switch(config-bgp-l2vpn-evpn) # no neighbor 2001:0db8:85a3::8a2e:0370:7334 route-reflector-client
```

neighbor send-community

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} send-community [standard | extended]
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} send-community [standard | extended]
```

Description

This command allows community values to be sent to a specific neighbor. When this command is configured for the peer-group, then all the neighbors that are part of peer-group will send the community values to the peers. The parameters `standard` and `extended` send only the respective community numbers. When the command is issued without either of these parameters, both standard and extended communities will be sent to the neighbor.

The `no` form of this command will not allow the neighbor to send community values to the specific neighbors that are part of peer-group.

Command context

```
config-bgp-ipv4-uc
```

or

```
config-bgp-ipv6-uc
```

or

```
config-bgp-l2vpn-evpn
```

Parameters

<IP-ADDRESS>

Specifies the neighbor IP address in the IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or in the IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.



NOTE: IPv6 MP-BGP peering must not be used for L2VPN EVPN address family, because VXLAN tunnel interface does not support IPv6 addresses.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 send-community standard
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 send-community standard
```

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 send-community extended
switch(config-bgp-ipv4-uc) # no neighbor 1.1.1.1 send-community
```

```
switch(config-bgp-ipv4-uc) # neighbor PG send-community standard
switch(config-bgp-ipv4-uc) # no neighbor PG send-community standard
```

```
switch(config-bgp-ipv4-uc) # neighbor PG send-community extended
switch(config-bgp-ipv4-uc) # no neighbor PG send-community
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 send-community standard
switch(config-bgp-l2vpn-evpn) # no neighbor 1.1.1.1 send-community standard
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 1.1.1.1 send-community extended
switch(config-bgp-l2vpn-evpn) # no neighbor 1.1.1.1 send-community
```

```
switch(config-bgp-l2vpn-evpn) # neighbor PG send-community standard
switch(config-bgp-l2vpn-evpn) # no neighbor PG send-community standard
```

```
switch(config-bgp-l2vpn-evpn) # neighbor PG send-community extended
switch(config-bgp-l2vpn-evpn) # no neighbor PG send-community
```

```
switch(config-bgp-l2vpn-evpn) # neighbor 2001:0db8:85a3::8a2e:0370:7334 send-community extended
switch(config-bgp-l2vpn-evpn) # no neighbor 2001:0db8:85a3::8a2e:0370:7334 send-community
```

neighbor shutdown

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} shutdown
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} shutdown
```

Description

This command disables the peer session, terminates any active session for the specified neighbor or peer group, and removes all associated routing information. This action can cause the sudden termination of many peering sessions.

The `no` form of this command enables the peer session for the specified neighbor.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Usage

Sessions are gracefully shut down when graceful-shutdown is enabled. Enter the `neighbor graceful-shutdown` command to enable graceful-shutdown. If graceful-shutdown is configured without delay or local-preference, the default values are used.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 shutdown
switch(config-bgp) # no neighbor 1.1.1.1 shutdown
```

```
switch(config-bgp) # neighbor pg shutdown
switch(config-bgp) # no neighbor pg shutdown
```

neighbor soft-reconfiguration inbound

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} soft-reconfiguration inbound  
  
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} soft-reconfiguration inbound
```

Description

This command enables you to generate inbound updates from a neighbor and change and activate BGP policies without clearing the BGP session. Changes in BGP policies require the BGP session to be cleared which can have a large negative impact on network operations.

The `no` form of this command disables this setting.

Command context

```
config-bgp-ipv4-uc  
or  
config-bgp-ipv6-uc
```

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- To perform inbound soft reconfiguration, the BGP speaker must store all received route updates, regardless of the current inbound policy.
- When inbound soft reconfiguration is enabled, the stored updates are processed by the new policy configuration to create new inbound updates.

Examples

```
switch(config-bgp-ipv4-uc) # neighbor 1.1.1.1 soft-reconfiguration inbound
```

neighbor timers

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} timers <KEEPALIVE> <HOLDTIME>  
  
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} timers <KEEPALIVE> <HOLDTIME>
```

Description

This command sets the timers for a specific BGP neighbor or peer group. When the timer is applied to peer-group then all the neighbors that are part of peer-group will inherit the value configured.

The neighbor connection must be reset using `clear ip bgp <NEIGHBOR-IP-ADDRESS>` to allow this configuration to take effect.

The `no` form of this command clears the timers for a specific BGP neighbor or peer group.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

<KEEPALIVE>

Specifies the Keep-Alive timer value for the neighbor. Default: 60 seconds. Range: 0-65535.

<HOLDTIME>

Specifies the Hold-timer value. Default: 180 seconds. Range: 0-65535.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp)# neighbor 1.1.1.1 timers 120 360
switch(config-bgp)# no neighbor 1.1.1.1 timers
```

```
switch(config-bgp)# neighbor pg timers 120 360
switch(config-bgp)# no neighbor pg timers
```

neighbor ttl-security-hops

Syntax

```
neighbor {<IP-ADDRESS> | <PEER-GROUP-NAME>} ttl-security-hops <HOP-COUNT>
```

```
no neighbor {<IP-ADDRESS> | <PEER-GROUP-NAME>} ttl-security-hops <HOP-COUNT>
```

Description

This command enables BGP to establish connection with external peers residing on networks that are not directly connected. By enabling this feature, the received TTL from a BGP peer is compared with the difference "255 - hop-count". BGP messages coming with a TTL less than this value are not accepted. BGP peering will not be established if the TTL in the session establishment is received with a lower value. Also, by enabling this feature the router will send BGP packets with TTL value of 255 to the neighbor. For a neighbor, either TTL security or `ebgp-multihop` can be configured, not both together. If there are multiple paths to reach the node, then the hop count should be configured considering the longest route.

The `no` form of this command disables the peer ttl-security-hop feature.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

<HOP-COUNT>

Specifies the hop count to reach the neighbor for the eBGP session. Range: 1-255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 ttl-security-hops 10  
switch(config-bgp) # no neighbor 1.1.1.1 ttl-security-hops
```

```
switch(config-bgp) # neighbor pg ttl-security-hops 5  
switch(config-bgp) # no neighbor pg ttl-security-hops
```

neighbor update-source

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>}  
    update-source {<IPv4>|<IPv6> | loopback <NUMBER>}  
  
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>}  
    update-source [<IPv4>|<IPv6> | loopback <NUMBER>]
```

Description

This command specifies the source address to reach the neighbor.

An iBGP connection can occur as long as there is a TCP/IP path between the routers. If multiple paths exist between the iBGP routers, using a loopback interface as the neighbor address can add stability to the network. With this command, stability can be achieved by providing the loopback interface address as the source address of the TCP/IP session.

The **no** form of this command negates the route updates of the neighbor.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

<IPV4>

Specifies an interface by IPv4 address.

<IPV6>

Specifies an interface by IPv6 address.

loopback <NUMBER>

Specifies a loopback interface number.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 update-source loopback 1
switch(config-bgp) # no neighbor 1.1.1.1 update-source
```

```
switch(config-bgp) # neighbor PG update-source loopback 1
switch(config-bgp) # no neighbor PG update-source
```

neighbor weight

Syntax

```
neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} weight <WEIGHT-VALUE>
```

```
no neighbor {<IP-ADDRESS>|<PEER-GROUP-NAME>} weight <WEIGHT-VALUE>
```

Description

This command assigns a weight to a neighbor connection. When the weight is applied to a peer-group then all the neighbors that are part of the peer-group will inherit the value configured.

The **no** form of this command removes a weight assignment.

Command context

config-bgp

Parameters

<IP-ADDRESS>

Specifies an IP address.

<PEER-GROUP-NAME>

Specifies a peer group.

<WEIGHT-VALUE>

Specifies the weight to be associated with the routes received from the neighbor. Range: 0-65535.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config-bgp) # neighbor 1.1.1.1 weight 500
switch(config-bgp) # no neighbor 1.1.1.1 weight
```

```
switch(config-bgp) # neighbor pg weight 600
switch(config-bgp) # no neighbor pg weight
```

network

Syntax

```
network { [<IPV4/MASK> | <IPV6/MASK>] } [route-map <ROUTE-MAP-NAME>]
```

```
no network { [<IPV4/MASK> | <IPV6/MASK>] } [route-map <ROUTE-MAP-NAME>]
```

Description

This command specifies the networks to be advertised by the Border Gateway Protocol (BGP) routing processes.

The `no` form of this command removes an entry from the routing table.

Command context

```
config-bgp-ipv4-uc
```

Parameters

<IPV4/MASK>

Specifies the IPv4 network with mask. For example: 1.1.1.1/24

<IPV6/MASK>

Specifies the IPv6 network with mask. For example: 2001:0db8:85a3::8a2e:0370:7334/24

route-map <ROUTE-MAP-NAME>

Optional parameter. Specifies a route map to apply to the prefixes advertised by this specific network statement.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- This command is used to advertise prefixes currently installed in the routing table into the BGP table.
- Use the `route-map` keyword to apply the specified route map to network advertisements. The mask length as configured in the network statement must match the mask length of prefixes in the routing table.

Examples

```
switch(config-bgp-ipv4-uc) # network 11.11.11.0/24  
switch(config-bgp-ipv4-uc) # no network 11.11.11.0/24
```

```
switch(config-bgp-ipv6-uc) # network 2001:0db8:85a3::8a2e:0370:7334/24  
switch(config-bgp-ipv6-uc) # no network 2001:0db8:85a3::8a2e:0370:7334/24
```

redistribute

Syntax

```
redistribute {connected|ospf|static|local loopback} [route-map <ROUTE-MAP-NAME>]
```

```
no redistribute {connected|ospf|static|local loopback} [route-map <ROUTE-MAP-NAME>]
```


Description

This command specifies routes to import into BGP. This command causes routes from the specified protocol to be considered for redistribution into BGP.

The `no` form of this command specifies no redistribution into BGP.

Command context

```
config-bgp-ipv4-uc
```

or

```
config-bgp-ipv6-uc
```

Parameters

{connected|ospf|static|local loopback}

Specifies a redistribution protocol name.

- `connected` - Redistributes directly attached networks (directly attached to the subnet or host).
- `ospf` - Redistributes Open Shortest Path First (OSPFv2) routes.
- `static` - Redistributes statically configured routes.
- `local loopback` - Redistributes local routes on loopback interfaces.

route-map <ROUTE-MAP-NAME>

Optional. Specifies a route map to match for redistribution.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- If a route map is specified, then routes that pass the match clause specified in the route map will be imported into the BGP peer Routing Information Base (RIB).
- Route-maps must be configured prior to being referenced in redistribution statements.
- Redistribute connected is required to redistribute connected subnet even if redistribute local loopback is already configured.

Examples

Redistribute directly attached networks:

```
switch(config-bgp-ipv4-uc) # redistribute connected  
switch(config-bgp-ipv4-uc) # no redistribute connected
```

Redistributing local routes on loopback interfaces:

```
switch(config-bgp-ipv4-uc) # redistribute local loopback  
switch(config-bgp-ipv4-uc) # no redistribute local loopback
```

router bgp

Syntax

```
router bgp <AS-NUMBER>
```

```
no router bgp <AS-NUMBER>
```

Description

This command configures the BGP instance on the router, configures the AS (Autonomous System) the router belongs to, and enters into the BGP router configuration mode. Only a single BGP AS number can be assigned for the entire system.

The `no` form of the command deletes the BGP instance from the router.

Command context

```
config
```

Parameters

AS-NUMBER

Required: Specifies a 4-byte AS number in the range 1-4294967295 in integer format or from 0.1-65535.65535 in dotted format.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring the BGP instance with the AS number:

```
switch(config)# router bgp 100
```

Deleting BGP configurations:

```
switch(config)# no router bgp 100  
This will delete all BGP configurations on this device.  
Continue (y/n)?
```

show bgp

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast  
    | all } [vsx-peer]
```

```
show bgp l2vpn evpn
```

Description

This command shows entries in the BGP routing table.

Command context

Operator (>) or Manager (#)

Parameters

ipv4

Selects the IPv4 address family.

ipv6

Selects the IPv6 address family.

unicast

The subaddress family identifier.

vrf <VRF-NAME>

Select to display information by VRFs by specifying the VRF name.

all

Select to display the BGP summary information for all VRFs and address-families.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing BGP routing table information for VRF 1 IPv4 unicast:

```
switch# show bgp vrf v1 ipv4 unicast
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : v1
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	65534.65535 3.4 18.54934 3574.8570 5.6 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing BGP routing table information for L2VPN EVPN:

```
switch# show bgp l2vpn evpn

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Router-ID not configured
```

Network	Nexthop	Metric	LocPrf	Weight	Path
---------	---------	--------	--------	--------	------

```

Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[] 1.1.1.20 0 100 32768 i
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[] 1.1.1.10 0 100 0 i
*> [3]:[0]:[1.1.1.1] 0.0.0.0 0 100 0 ?
Total number of entries 3

```

show bgp <PREFIX>

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast <A.B.C.D/M> |
    ipv6 unicast <X::Y/M>} [vsx-peer]
```

```
show bgp l2vpn evpn <VXLAN-NETWORK-ID>
```

Description

This command displays entries in the BGP routing table that are part of the specified network.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast <A.B.C.D/M>

Shows the information for an IPv4 unicast family with an IP prefix (network/length such as 35.0.0.0/8) in the BGP routing table to display.

ipv6 unicast <X::Y/M>

Shows the information for an IPv6 unicast family an IPv6 prefix in the BGP routing table to display.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing the entries in the BGP routing table that are part of an IPv4 unicast network:

```
switch# show bgp ipv4 unicast 10.0.0.0/16
```

```

VRF : default
BGP Local AS 2      BGP Router-ID 1.1.1.2

Network      : 10.0.0.0/16      Nexthop      : 1.1.1.1
Peer         : 1.1.1.1         Origin       : IGP
Metric       : 0               Local Pref   : 100
Weight       : 0               Calc. Local Pref : 100
Best         : Yes             Valid        : Yes
Type         : external        Stale        : No

```

```

Originator ID      : 0.0.0.0                Path ID           : 0
Aggregator ID      :
Aggregator AS       :
Atomic Aggregate    :
RFD Flaps           : 0                    RFD Penalty       : 0

AS-Path            : 1
Cluster List       :
Communities         : 50:100,50:101,50:102,50:103,50:104,50:105,50:106,50:107,50:108,50:109,50:110,50:1
Extd. Communities  :

```

Showing the entries in the BGP routing table that are part of L2VPN EVPN:

```

switch# show bgp l2vpn evpn vni 30000
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Router-ID not configured

```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)					
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[]	1.1.1.20	0	100	32768	i
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[]	1.1.1.10	0	100	0	i
*> [3]:[0]:[1.1.1.1]	0.0.0.0	0	100	0	?
Total number of entries 3					

show bgp community

Syntax

```

show bgp [{ipv4 | ipv6 | ipv4 {vrf <VRF-NAME>}} unicast]
        community [<VALUE> | <TYPE>] [vsx-peer]

```

Description

This command shows routes that belong to BGP communities. Optionally you can specify displaying information by a specific community or by VRF.

Command context

Operator (>) or Manager (#)

Parameters

ipv4

Shows the information for an IPv4 address family.

ipv6

Shows the information for an IPv6 address family.

unicast

Shows the information for a subaddress family identifier.

ipv4 vrf <VRF-NAME>

Shows the information for a specified VRF.

<VALUE>

Shows the information for a community number. Specify the information in aa:nn format.

<TYPE>

Shows a specified community type. Select the following well-known communities, as well as others:

internet

Advertise the prefix to all BGP neighbors.

local-as

Do not advertise the prefix outside the sub-AS.

no-advertise

Do not advertise the prefix to any BGP neighbors.

no-export

Do not advertise the prefix to any eBGP neighbors.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing information for routes that belong to all BGP communities:

```
switch# show bgp ipv4 unicast community
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing information for routes that belong to the 200:20 BGP community number:

```
switch# show bgp ipv4 unicast community 200:20
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?

```
*>e 100.0.3.0/24      9.0.0.2      0      100      0      200 ?
*ae 100.0.3.0/24      9.0.0.3      0      100      0      200 ?
Total number of entries 6
```

Showing information for routes that belong to the Internet BGP community type:

```
switch# show bgp ipv4 unicast community internet
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing information for routes that belong to the local-as BGP community type:

```
switch# show bgp ipv4 unicast community local-as
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing information for routes that belong to the no-advertise BGP community type:

```
switch# show bgp ipv4 unicast community no-advertise
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.2	0	100	0	200 ?

Total number of entries 6

Showing information for routes that belong to the no-export BGP community type:

```
switch# show bgp ipv4 unicast community no-export
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete
```

VRF : default
Local Router-ID 9.0.0.1

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing information for routes that belong to the gshut BGP community type:

```
switch# show bgp ipv4 unicast community gshut
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

VRF : default
Local Router-ID 1.1.1.2

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 1.1.1.0/24	10.1.1.2	0	0	0	2 i

Total number of entries 1

```
switch#
switch# show bgp ipv6 unicast community gshut
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

VRF : default
Local Router-ID 1.1.1.2

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 1::/64	10::2				
	fe80::98f2:b300:1368:e882	0	0	0	2 i

Total number of entries 1

show bgp flap-statistics

Syntax

```
show bgp {vrf <VRF_NAME> | all-vrf} [{ipv4 unicast | ipv6 unicast | all}] flap-statistics
```

Description

Displays all the flapped and suppressed routes.

Authority

Administrators or local user group members with execution rights for this command.

Usage

Status of the route with dampening enabled:

- If the route is available, the history flag is unset.
- If route has been flapping, is not suppressed and is withdrawn; the state of the route is h
- If route is currently available but is suppressed due to dampening, the state of the route is d
- If the route is unsuppressed and currently withdrawn, the state of the route is h

Examples

Showing all the flapped and suppressed routes:

```
switch# show bgp all
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : default
Local Router-ID 1.1.1.1

Address-family : IPv4 Unicast
-----
      Network          Nexthop          Metric      LocPrf      Weight Path
*>i 2.2.2.0/24          2.2.2.2            0           100         0        ?
*>i 11.1.1.0/24          2.2.2.2            0           100         0        ?
*ai 11.1.1.0/24         2.2.2.3            0           100         0        ?
Total number of entries 3

Address-family : IPv6 Unicast
-----
      Network          Nexthop          Metric      LocPrf      Weight Path
Total number of entries 0
```

```
switch# show bgp ipv4 unicast flap-statistics
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : default
Local Router-ID 20.0.0.1

      Network          Nexthop          Flaps          Reuse          Path
*>e 2.2.2.0/24          20.0.0.2          1              00h:00m:00s    300 ?
de 3.3.3.0/24          20.0.0.2          2              00h:29m:31s    300 ?
Total number of entries 2
```

show bgp neighbors

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast | all}
      neighbors [vsx-peer]
```

```
show bgp l2vpn evpn neighbors
```

Description

This command shows information about BGP and TCP connections to neighbors. If neighbors are member of a peer-group, the command shows the configured values inherited from the peer-group. The configured values are postfixed with a caret (^) for inherited values.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 6300, 6400, 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing all information about BGP and TCP connections to neighbors:

```
switch# show bgp all neighbors
Codes: ^ Inherited from peer-group
VRF : default
```

```
BGP Neighbor 10.1.1.2 (Internal)
  Description          :
  Peer-group           :

  Remote Router Id     : 10.1.1.2          Local Router Id   : 1.0.0.1
  Remote AS             : 1                Local AS          : 1
  Remote Port          : 0                Local Port        : 0
  State                 : Idle              Admin Status     : Up
  Conn. Established    : 0                Conn. Dropped     : 0
  Passive              : No               Update-Source      :
  Cfg. Hold Time       : 180              Cfg. Keep Alive   : 60
  Neg. Hold Time       : 0                Neg. Keep Alive    : 0
```

```

Up/Down Time      : 00h:00m:00s      Alt. Local-AS      : 0
Local-AS Prepend  : No
BFD                : Disabled
Password          :
Last Err Sent     : No Error
Last SubErr Sent  : No Error
Last Err Rcvd    : No Error
Last SubErr Rcvd  : No Error

Graceful-Restart   : Enabled          Gr. Restart Time   : 120
Gr. Stalepath Time : 150              Remove Private-AS  : No
TTL                : 255              Local Cluster-ID   :
Weight             : 0                Fall-over          : No
Confederation-Peers : No

```

Message statistics	Sent	Rcvd
-----	-----	-----
Open	0	0
Notification	0	0
Updates	0	0
Keepalives	0	0
Route Refresh	0	0
Total	0	0

Capability	Advertised	Received
-----	-----	-----
Route Refresh	Yes	No
Graceful Restart	Yes	No
Add-Path	No	No
Four Octet ASN	Yes	No
Address family IPv4 Unicast	Yes	No
Address family IPv6 Unicast	No	No
Address family L2VPN EVPN	No	No

Address Family : IPv4 Unicast

Rt. Reflect. Client	: No	Send Community	:
Allow-AS in	: 0	Advt. Interval	: 30
Max. Prefix	: 64000	Soft Reconfig In	:
Nexthop-Self	:	Default-Originate	:
Cfg. Add-Path	: Send		
Neg. Add-Path	: Send		

```

Routemap In      :
Routemap Out     :
ORF Type         : Prefix-list
ORF Capability    : Receive

```

Address Family : IPv6 Unicast

```

ORF Type         : Prefix-list
ORF Capability    : Receive

```

```

Cfg. Add-Path    : Send
Neg. Add-Path    : Send

```

Address Family : L2VPN EVPN

Showing information about L2VPN EVPN connections to neighbors:

switch# **show bgp l2vpn evpn neighbors**

Codes: ^ Inherited from peer-group

VRF : default

BGP Neighbor 10.1.1.2 (Internal)

Description	:		
Peer-group	:		
Remote Router Id	:	10.1.1.2	Local Router Id : 10.1.1.1
Remote AS	:	1	Local AS : 1
Remote Port	:	179	Local Port : 56008
State	:	Established	Admin Status : Up
Conn. Established	:	1	Conn. Dropped : 0
Passive	:	No	Update-Source :
Cfg. Hold Time	:	180	Cfg. Keep Alive : 60
Neg. Hold Time	:	180	Neg. Keep Alive : 60
Up/Down Time	:	00m:01w:03d	Alt. Local-AS : 0
Local-AS Prepend	:	No	
BFD	:	Disabled	
Password	:		
Last Err Sent	:	No Error	
Last SubErr Sent	:	No Error	
Last Err Rcvd	:	No Error	
Last SubErr Rcvd	:	No Error	
Graceful-Restart	:	Enabled	Gr. Restart Time : 120
Gr. Stalepath Time	:	150	Remove Private-AS : No
TTL	:	255	Local Cluster-ID :
Weight	:	0	Fall-over : No
Confederation-Peers	:	No	

Message statistics	Sent	Rcvd
-----	-----	-----
Open	1	1
Notification	0	0
Updates	3	2
Keepalives	17995	18009
Route Refresh	0	0
Total	17999	18012

Capability	Advertised	Received
-----	-----	-----
Route Refresh	Yes	Yes
Graceful Restart	Yes	Yes
Four Octet ASN	Yes	Yes
Add-Path	Yes	No
Address family IPv4 Unicast	Yes	Yes
Address family IPv6 Unicast	Yes	Yes
Address family L2VPN EVPN	Yes	Yes

Address Family : L2VPN EVPN

Rt. Reflect. Client	:	No	Send Community	:	extended
Allow-AS in	:	0	Advt. Interval	:	30
Max. Prefix	:	32768	Soft Reconfig In	:	
Nexthop-Self	:		Default-Originate	:	
Routemap In	:				
Routemap Out	:				

show bgp neighbor advertised-routes

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast  
    | l2vpn evpn | all} neighbors <IP-ADDRESS>  
    advertised-routes [vsx-peer]
```

```
show bgp l2vpn evpn neighbors <IP-ADDRESS> advertised-routes
```

Description

Shows all routes that have been advertised to the specified neighbor.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter applies only to 8325 and 8360 series switches.

all

Shows the information for all address families and subaddress families.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

<IP-ADDRESS>

Shows the information for a neighbor IP address.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing routes that have been advertised to the specified IPv4 unicast neighbor:

```
switch# show bgp ipv4 unicast neighbors 9.0.0.1 advertised-routes  
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,  
               i internal, e external S Stale, R Removed  
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
VRF : default  
Local Router-ID 9.0.0.2
```

Network	Nexthop	Metric	LocPrf	Weight	Path
---------	---------	--------	--------	--------	------

```

-----
*>e 9.0.0.0/24          9.0.0.2      0          0          0          200 65534.65535 3.4 18.54934 3574.8570 5.6 ?
*>e 100.0.0.0/24         9.0.0.2      0          0          0          200 ?
*>e 100.0.1.0/24         9.0.0.2      0          0          0          200 ?
*>e 100.0.2.0/24         9.0.0.2      0          0          0          200 ?
*>e 100.0.3.0/24         9.0.0.2      0          0          0          200 ?
Total number of entries 5

```

Showing routes that have been advertised to the specified L2VPN EVPN neighbor:

```

switch# show bgp l2vpn evpn neighbor 9.0.0.1 advertised-routes
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Local Router-ID 9.0.0.2

```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)					
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[]	1.1.1.20	0	100	32768	i
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[]	1.1.1.10	0	100	0	i
Total number of entries 2					

show bgp neighbor paths

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast
                             | all} neighbors <IP-ADDRESS> paths [vsx-peer]
```

```
show bgp l2vpn evpn neighbors <IP-ADDRESS> paths
```

Description

Shows autonomous system paths learned from the specified neighbor.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

<IP-ADDRESS>

Shows the information for a neighbor IP address.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter applies only to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing autonomous system paths learned from the specified IPv4 unicast neighbor:

```
switch# show bgp ipv4 unicast neighbors 192.168.12.2 paths
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Path
*>e 9.0.0.0/24	9.0.0.2	200 65534.65535 3.4 18.54934 3574.8570 5.6
*>e 100.0.0.0/24	9.0.0.2	200
*>e 100.0.1.0/24	9.0.0.2	200
*>e 100.0.2.0/24	9.0.0.2	200
*>e 100.0.3.0/24	9.0.0.2	200

Total number of entries 5

Showing autonomous system paths learned from the specified L2VPN EVPN neighbor:

```
switch# show bgp l2vpn evpn neighbors 192.168.12.1 paths

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Local Router-ID 9.0.0.2
```

Network	Nexthop	Path
Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)		
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[]	1.1.1.20	100
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[]	1.1.1.10	100

Total number of entries 2

show bgp neighbor received orf-prefix-list

Syntax

```
show bgp [vrf <VRF-NAME>]{ipv4 unicast | ipv6 unicast | all} neighbors <IP-ADDRESS> received orf-prefix-list
```

Description

Shows all the prefix list received from the specified neighbor.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

<IP-ADDRESS>

Shows the information for a neighbor IP address.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing received prefix list from the specified neighbor:

```
switch# show bgp ipv4 unicast neighbors A.B.C.D received orf-prefix-list
Address family: IPv4 Unicast

ip prefix-list 10.0.0.200: 4 entries
    seq 10 permit 28.119.16.0/24
    seq 15 deny 28.119.19.0/24
    seq 20 permit 28.119.17.0/24
Address family: IPv6 Unicast
ip prefix-list 10.0.0.200: 4 entries
    seq 30 permit 2000::/64
    seq 35 deny 3000::/64
    seq 40 permit 4000:0/64
```

show bgp neighbor received-routes

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast
    | all} neighbors <IP-ADDRESS> received-routes [vsx-peer]
```

```
show bgp l2vpn evpn neighbors <IP-ADDRESS> received-routes
```


Description

Shows received routes from the specified neighbor.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

<IP-ADDRESS>

Shows the information for a neighbor IP address.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing received routes from the specified IPv4 unicast neighbor:

```
switch# show bgp ipv4 unicast neighbors 192.168.12.1 received-routes
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : default
Local Router-ID 9.0.0.2
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	0	0	200 65534.65535 3.4 18.54934 3574.8570 5.6 ?
*>e 100.0.0.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	0	0	200 ?
*ae 100.0.3.0/24	9.0.0.2	0	0	0	200 ?

Total number of entries 6

Showing received routes from the specified L2VPN EVPN neighbor:

```
switch# show bgp l2vpn evpn neighbors 192.168.12.1 received-routes
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
```

```

        i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Local Router-ID 9.0.0.2

```

Network	NextHop	Metric	LocPrf	Weight	Path
Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)					
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[]	1.1.1.20	0	100	32768	i
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[]	1.1.1.10	0	100	0	i
Total number of entries 2					

show bgp neighbor routes

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast
    | all} neighbors <IP-ADDRESS> routes [vsx-peer]
```

```
show bgp l2vpn evpn neighbors <IP-ADDRESS>
```

Description

This command shows routes that are received and accepted from the specified neighbor.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

<IP-ADDRESS>

Shows the information for a neighbor IP address.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing all routes that are received and accepted from the specified neighbor:

```
switch# show bgp ipv4 unicast neighbors 9.0.0.2 routes
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	100	0	200 65534.65535 3.4 18.54934 3574.8570 5.6 ?
*>e 100.0.0.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	100	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	100	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

Showing 12 VPN EVPN routes that are received and accepted from the specified neighbor:

```
switch# show bgp l2vpn evpn neighbor 9.0.0.2 routes

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Local Router-id: 9.0.0.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 9.0.0.0/24	9.0.0.2	0	0	0	200 65534.65535 3.4 18.54934 3574.8570 5.6 ?
*>e 100.0.0.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.1.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.2.0/24	9.0.0.2	0	0	0	200 ?
*>e 100.0.3.0/24	9.0.0.2	0	0	0	200 ?
*ae 100.0.3.0/24	9.0.0.3	0	100	0	200 ?

Total number of entries 6

show bgp neighbors

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast | all}
      neighbors [vsx-peer]
```

```
show bgp l2vpn evpn neighbors
```

Description

This command shows information about BGP and TCP connections to neighbors. If neighbors are member of a peer-group, the command shows the configured values inherited from the peer-group. The configured values are postfixed with a caret (^) for inherited values.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter applies only to 8325, 8360, 6300 and 6400 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing all information about BGP and TCP connections to neighbors:

```
switch# show bgp all neighbors
Codes: ^ Inherited from peer-group
VRF : default

BGP Neighbor 10.1.1.2 (Internal)
  Description          :
  Peer-group           :

  Remote Router Id     : 10.1.1.2          Local Router Id    : 1.0.0.1
  Remote AS            : 1                 Local AS           : 1
  Remote Port          : 0                 Local Port         : 0
  State                : Idle              Admin Status      : Up
  Conn. Established    : 0                 Conn. Dropped      : 0
  Passive              : No                Update-Source      :
  Cfg. Hold Time       : 180                Cfg. Keep Alive    : 60
  Neg. Hold Time       : 0                  Neg. Keep Alive    : 0
  Up/Down Time         : 00h:00m:00s        Alt. Local-AS      : 0
  Local-AS Prepend     : No
  BFD                  : Disabled
  Password             :
  Last Err Sent        : No Error
  Last SubErr Sent     : No Error
  Last Err Rcvd        : No Error
  Last SubErr Rcvd     : No Error

  Graceful-Restart      : Enabled            Gr. Restart Time   : 120
  Gr. Stalepath Time    : 150                Remove Private-AS  : No
  TTL                   : 255                 Local Cluster-ID   :
  Weight                : 0                   Fall-over          : No

  Message statistics    Sent      Rcvd
  -----
  Open                  0         0
  Notification          0         0
  Updates               0         0
  Keepalives            0         0
  Route Refresh         0         0
```

```

Total                                0          0

Capability                           Advertised    Received
-----
Route Refresh                       Yes          No
Graceful Restart                     Yes          No
Four Octet ASN                       Yes          No
Address family IPv4 Unicast          Yes          No
Address family IPv6 Unicast          No           No
Address family L2VPN EVPN            No           No

Address Family : IPv4 Unicast
-----

Rt. Reflect. Client : No             Send Community      :
Allow-AS in          : 0              Advt. Interval      : 30
Max. Prefix          : 64000          Soft Reconfig In    :
Nexthop-Self         :                Default-Originate   :

Routemap In          :
Routemap Out         :
Address Family : IPv6 Unicast
-----

Address Family : L2VPN EVPN
-----

```

Showing information about L2VPN EVPN connections to neighbors:

```

switch# show bgp l2vpn evpn neighbors
Codes: ^ Inherited from peer-group

VRF : default

BGP Neighbor 10.1.1.2 (Internal)
Description          :
Peer-group           :

Remote Router Id     : 10.1.1.2      Local Router Id     : 10.1.1.1
Remote AS            : 1              Local AS            : 1
Remote Port          : 179            Local Port          : 56008
State                : Established    Admin Status        : Up
Conn. Established    : 1              Conn. Dropped       : 0
Passive              : No             Update-Source       :
Cfg. Hold Time       : 180            Cfg. Keep Alive     : 60
Neg. Hold Time       : 180            Neg. Keep Alive     : 60
Up/Down Time         : 00m:01w:03d    Alt. Local-AS       : 0
Local-AS Prepend     : No
BFD                  : Disabled
Password             :
Last Err Sent        : No Error
Last SubErr Sent     : No Error
Last Err Rcvd        : No Error
Last SubErr Rcvd     : No Error

Graceful-Restart      : Enabled        Gr. Restart Time    : 120
Gr. Stalepath Time   : 150            Remove Private-AS   : No
TTL                  : 255            Local Cluster-ID    :
Weight               : 0              Fall-over            : No

Message statistics    Sent      Rcvd
-----
Open                  1          1
Notification          0          0

```

Updates	3	2
Keepalives	17995	18009
Route Refresh	0	0
Total	17999	18012

Capability	Advertised	Received
Route Refresh	Yes	Yes
Graceful Restart	Yes	Yes
Four Octet ASN	Yes	Yes
Address family IPv4 Unicast	Yes	Yes
Address family IPv6 Unicast	Yes	Yes
Address family L2VPN EVPN	Yes	Yes

Address Family : L2VPN EVPN

Rt. Reflect. Client : No	Send Community : extended
Allow-AS in : 0	Advt. Interval : 30
Max. Prefix : 32768	Soft Reconfig In :
Nexthop-Self :	Default-Originate :
Routemap In :	
Routemap Out :	

show bgp paths

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast | all}
      paths [vsx-peer]
```

```
show bgp l2vpn evpn paths
```

Description

Shows received BGP path information in the database.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing received BGP path information from the specified IPv4 unicast neighbor:

```
switch# show bgp ipv4 unicast paths
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed, a additional-paths
VRF : default
Local Router-ID 9.0.0.1
```

Network	Nexthop	PathID	Path
*>e 9.0.0.0/24	9.0.0.2	0	200 65534.65535 3.4 18.54934 3574.8570 5.6
*>e 100.0.0.0/24	9.0.0.2	0	200
*>e 100.0.1.0/24	9.0.0.2	0	200
*>e 100.0.2.0/24	9.0.0.2	0	200
*>e 100.0.3.0/24	9.0.0.2	10	200
*ae 100.0.3.0/24	9.0.0.3	5	200

Total number of entries 6

Showing received BGP path information from the specified L2VPN EVPN neighbor:

```
switch# show bgp l2vpn evpn paths

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]

VRF : default
Local Router-ID 9.0.0.2
```

Network	Nexthop	Path
Route Distinguisher: 10.1.1.54:32967 (L2VNI 30000)		
*> [2]:[0]:[0]:[00:06:f6:3f:e3:c1]:[]	1.1.1.20	100
*> [2]:[0]:[0]:[8c:60:4f:f2:f5:41]:[]	1.1.1.10	100

Total number of entries 2

show bgp peer-group summary

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast |
                             ipv6 unicast | all} peer-group <PEER-GROUP-NAME>
                             summary [vsx-peer]
```

```
show bgp l2vpn evpn peer-group <PEER-GROUP-NAME> summary
```

Description

This command shows the peer-group information in the database.

Command context

Operator (>) or Manager (#)

Parameters

vrf <VRF-NAME>

Shows the information for a specified VRF.

ipv4 unicast

Shows the information for an IPv4 unicast address family.

ipv6 unicast

Shows the information for an IPv6 unicast address family.

all

Shows the information for all address families and subaddress families.

<PEER-GROUP-NAME>

Shows the information for the BGP peer-group for the BGP instance.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing the information from IPv4 unicast address families in pg_name1 peer-group:

```
switch# show bgp ipv4 unicast peer-group pg_name1 summary
VRF : default
BGP Peer-Group Summary
=====
Local AS           : 1           BGP Router Identifier : 2.2.2.2
Peers              : 1           Log Neighbor Changes  : No
Cfg. Hold Time     : 180        Cfg. Keep Alive       : 60
Confederation Id   : 0

Neighbor           Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.1           1         8        10       00h:00m:58s  Established Up
```

Showing the information from all address families in pg_name1 peer-group:

```
switch# show bgp all unicast peer-group pg_name1 summary
VRF : default
BGP Peer-Group Summary
=====
Local AS           : 1           BGP Router Identifier : 2.2.2.2
Peers              : 1           Log Neighbor Changes  : No
Cfg. Hold Time     : 180        Cfg. Keep Alive       : 60
Confederation Id   : 0
```



```

For address family: IPv4 Unicast
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.1      1          8        10       00h:00m:58s   Established Up
For address family: IPv6 Unicast
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.1      1          8        10       00h:00m:58s   Established Up
For address family: L2VPN EVPN
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.1      1          8        10       00h:00m:58s   Established Up

```

Showing the information from L2VPN EVPN address families in pg_name1 peer-group:

```

switch# show bgp l2vpn evpn peer-group pg_name1 summary
VRF : default
BGP Peer-Group Summary
=====
Local AS           : 1          BGP Router Identifier : 2.2.2.2
Peers              : 1          Log Neighbor Changes  : No
Cfg. Hold Time     : 180       Cfg. Keep Alive       : 60
Confederation Id   : 0

Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.1      1          8        10       00h:00m:58s   Established Up

```

show bgp summary

Syntax

```
show bgp [vrf <VRF-NAME>] {ipv4 unicast | ipv6 unicast | all}
      summary [vsx-peer]
```

```
show bgp l2vpn evpn summary
```

Description

This command shows a summary of the status of Border Gateway Protocol (BGP) connections.

Command context

Operator (>) or Manager (#)

Parameters

ipv4 unicast

Selects to display the BGP summary information for the IPv4 subaddress family identifier.

ipv6 unicast

Selects to display the BGP summary information for the IPv6 subaddress family identifier.

all

Selects to display the BGP summary information for all VRFs and address-families.

vrf <VRF-NAME>

Selects to display information by VRFs by specifying the VRF name.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

l2vpn evpn

Shows the information for L2VPN EVPN address family. This parameter only applies to 8325 and 8360 series switches.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing BGP summary information for all address-families:

```
switch(config-bgp)# show bgp all summary
VRF : default
BGP Summary
  Local AS           : 100           BGP Router Identifier : 9.0.0.1
  Peers              : 1             Log Neighbor Changes  : No
  Cfg. Hold Time     : 180          Cfg. Keep Alive       : 60
  Confederation Id   : 0

Address-family : IPv4 Unicast
-----
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
9.0.0.2       200        25       23       00h:17m:50s  Established Up

Address-family : IPv6 Unicast
-----
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
9000::2       200        25       23       00h:17m:50s  Established Up

Address-family : L2VPN EVPN
-----
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
10.0.0.2      200        25       23       00h:17m:50s  Established Up

VRF : v1
BGP Summary
  Local AS           : 100           BGP Router Identifier : 9.0.0.1
  Peers              : 1             Log Neighbor Changes  : No
  Cfg. Hold Time     : 180          Cfg. Keep Alive       : 60
  Confederation Id   : 0

Address-family : IPv4 Unicast
-----
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
9.0.0.2       200        25       23       00h:17m:50s  Established Up

Address-family : IPv6 Unicast
-----
Neighbor      Remote-AS  MsgRcvd  MsgSent  Up/Down Time  State      AdminStatus
9000::2       200        25       23       00h:17m:50s  Established Up
```

Showing BGP summary information for a specific VRF for IPv4 unicast network:

```
switch(config-bgp)# show bgp ipv4 unicast vrf v1 summary
VRF : v1
BGP Summary
  Local AS           : 100           BGP Router Identifier : 9.0.0.1
  Peers              : 1             Log Neighbor Changes  : No
  Cfg. Hold Time     : 180          Cfg. Keep Alive       : 60
  Confederation Id   : 0
```

Neighbor	Remote-AS	MsgRcvd	MsgSent	Up/Down	Time	State	AdminStatus
9.0.0.2	200	25	23	00h:17m:50s		Established	Up

Showing BGP summary information for L2VPN EVPN:

```
switch(config-bgp)# do show bgp l2vpn evpn summary
```

VRF : default

BGP Summary

Local AS	: 100	BGP Router Identifier	: 9.0.0.1
Peers	: 1	Log Neighbor Changes	: No
Cfg. Hold Time	: 180	Cfg. Keep Alive	: 60
Confederation Id	: 0		

Neighbor	Remote-AS	MsgRcvd	MsgSent	Up/Down	Time	State	AdminStatus
10.0.0.2	200	25	23	00h:17m:50s		Established	Up

show bgp l2vpn evpn vni route-type

Syntax

```
show bgp l2vpn evpn vni <VNI-Value> route-type <ROUTE-TYPE-Value>
```

Description

Shows the BGP L2VPN information for the particular EVPN VNI and routes type.

Command context

Operator (>) or Manager (#)

Parameter

<VNI-Value>

Specifies the VNI.

<ROUTE-TYPE-Value>

Specifies the routes filtered by NLRI route type.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing BGP L2VPN information for the particular EVPN VNI and route type:

```
switch# show bgp l2vpn evpn vni 30000 route-type 5
```

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
i internal, e external S Stale, R Removed

Origin codes: i - IGP, e - EGP, ? - incomplete

EVPN Route-Type 5 prefix: [5]:[ESI]:[EthTag]:[IPAddrLen]:[IPAddr]

VRF : default

Router-ID not configured

Network	Nexthop	Metric	LocPrf	Weight	Path

Route Distinguisher: 1:100 (L3VNI 10000)					
*>i [5]:[0]:[0]:[24]:[32.32.32.0]	3.3.3.3	0	100	0	?
*> [5]:[0]:[0]:[24]:[52.52.52.0]	1.1.1.1	0	100	0	?
*>i [5]:[0]:[0]:[64]:[aaa::]	3.3.3.3	0	100	0	?

Total number of entries 3

show bgp l2vpn evpn vtep

Syntax

```
show bgp l2vpn evpn vtep <IP-address>
```

Description

Shows the BGP L2VPN information for the particular EVPN VTEP IP address.

Command context

Operator (>) or Manager (#)

Parameter

<IP-address>

Specifies the VTEP IP address.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing BGP L2VPN information for the particular EVPN VTEP IP:

```
switch# show bgp l2vpn evpn vtep 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]
EVPN Route-Type 5 prefix: [5]:[ESI]:[EthTag]:[IPAddrLen]:[IPAddr]
VRF : default
Local Router-ID 2.2.2.2
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 1.1.1.1:2 (L2VNI 2)					
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[10.1.1.10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[1000::10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[10.1.1.1]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[]	1.1.1.1	0	100	0	?
*>i [3]:[0]:[1.1.1.1]	1.1.1.1	0	100	0	?
Route Distinguisher: 1.1.1.1:2 (L3VNI 10000)					
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[10.1.1.10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[1000::10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[10.1.1.1]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[]	1.1.1.1	0	100	0	?
Route Distinguisher: 1:100 (L3VNI 10000)					
*>i [5]:[0]:[0]:[24]:[10.1.1.0]	1.1.1.1	0	100	0	?
*>i [5]:[0]:[0]:[64]:[1000::]	1.1.1.1	0	100	0	?
Total number of entries 11					

show bgp l2vpn evpn vtep route-type

Syntax

```
show bgp l2vpn evpn vtep <IP-address> route-type <ROUTE-TYPE-Value>
```

Description

Shows the BGP L2VPN information for the particular EVPN VTEP IP address and routes type.

Command context

Operator (>) or Manager (#)

Parameter

<IP-address>

Specifies the VTEP IP address.

<ROUTE-TYPE-Value>

Specifies the routes filtered by NLRI route type.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing BGP L2VPN information for the particular EVPN VTEP and route type:

```
switch# show bgp l2vpn evpn vtep 1.1.1.1 route-type 5
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN Route-Type 5 prefix: [5]:[ESI]:[EthTag]:[IPAddrLen]:[IPAddr]

VRF : default
Router-ID not configured
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 1:100 (L3VNI 10000)					
*>i [5]:[0]:[0]:[24]:[32.32.32.0]	1.1.1.1	0	100	0	?
*> [5]:[0]:[0]:[24]:[52.52.52.0]	1.1.1.1	0	100	0	?
*>i [5]:[0]:[0]:[64]:[aaa::]	1.1.1.1	0	100	0	?

Total number of entries 3

show bgp l2vpn evpn vtep vni

Syntax

show bgp l2vpn evpn vtep <IP-address> vni <VNI-Value>

Description

Shows the BGP L2VPN information for the particular EVPN VTEP IP address and VNI.

Command context

Operator (>) or Manager (#)

Parameter

<IP-address>

Specifies the VTEP IP address.

<VNI-Value>

Specifies the VNI.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing BGP L2VPN information for the particular EVPN VTEP IP and VNI:

```
switch# show bgp l2vpn evpn vtep 1.1.1.1 vni 10000
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]
EVPN Route-Type 5 prefix: [5]:[ESI]:[EthTag]:[IPAddrLen]:[IPAddr]
VRF : default
Local Router-ID 2.2.2.2
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 1.1.1.1:2 (L3VNI 10000)					
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[10.1.1.10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:00:00:00:00:33]:[1000::10]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[10.1.1.1]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:15:1c]:[]	1.1.1.1	0	100	0	?
Route Distinguisher: 1:100 (L3VNI 10000)					
*>i [5]:[0]:[0]:[24]:[10.1.1.0]	1.1.1.1	0	100	0	?
*>i [5]:[0]:[0]:[64]:[1000::]	1.1.1.1	0	100	0	?
Total number of entries 6					

show bgp l2vpn evpn vtep vni route-type

Syntax

```
show bgp l2vpn evpn vtep <VTEP-ID> vni <VNI-Value> route-type <ROUTE-TYPE-Value>
```

Description

Shows the BGP L2VPN information for the particular EVPN VTEP, VNI, and router type.

Command context

Operator (>) or Manager (#)

Parameter

<VTEP-ID>

Specifies the VTEP.

<VNI-Value>

Specifies the VNI.

<ROUTE-TYPE-Value>

Specifies the router type.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing BGP L2VPN information for the particular EVPN VTEP, route type, and VNI:

```
switch# show bgp l2vpn evpn vtep 1.1.1.1 vni 10000 route-type 2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

EVPN Route-Type 2 prefix: [2]:[ESI]:[EthTag]:[MAC]:[OrigIP]
EVPN Route-Type 3 prefix: [3]:[EthTag]:[OrigIP]
EVPN Route-Type 5 prefix: [5]:[ESI]:[EthTag]:[IPAddrLen]:[IPAddr]
VRF : default
```

Local Router-ID 2.2.2.2

Network	Nexthop	Metric	LocPrf	Weight	Path

Route Distinguisher: 1.1.1.1:2 (L3VNI 10000)					
*>i [2]:[0]:[0]:[00:50:56:96:7d:03]:[10.1.1.1]	1.1.1.1	0	100	0	?
*>i [2]:[0]:[0]:[00:50:56:96:7d:03]:[]	1.1.1.1	0	100	0	?
Total number of entries 3					

show running-config bgp

Syntax

```
show running-config bgp [vsx-peer]
```

Description

This command shows all configured BGP commands.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

EVPN only applies to 8325, 8360, 6300 and 6400 series switches.

```
switch# show running-config bgp
router bgp 65534.65535
  bgp asnotation dotted
  network 2.2.2.0/24
  neighbor 1.1.1.2 remote-as 65533.65535
  address-family ipv4 unicast
    neighbor 1.1.1.2 activate
    neighbor 1.1.1.2 route-map A out
  vrf v1
  address-family l2vpn evpn
    neighbor 1.1.1.2 activate
    neighbor 1.1.1.2 send-community extended
  exit-address-family
```

timers bgp

Syntax

```
timers bgp <KEEPALIVE> <HOLDTIME>
```

```
no timers bgp <KEEPALIVE> <HOLDTIME>
```

Description

The command adjusts BGP network timers.

The `no` form of this command resets the BGP timers to defaults of 60 seconds for the keepalive timer and 180 seconds for the holdtime timer.

Command context

`config-bgp`

Parameters

<KEEPALIVE>

Sets the value for keepalive timer. Default: 60 seconds. Range: 0-65535.

<HOLDTIME>

Sets the value for holdtime timer. Default: 180 seconds. Range: 0-65535.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- The keepalive timer is the number of seconds a BGP peer waits for a keep-alive message from a BGP peer before deciding the connection is down.
The holdtime timer is the number of seconds a BGP peer waits after not receiving a keepalive, update, or notification message before declaring that a connection with BGP peer is down.
- When a session is started, BGP negotiates holdtime with the neighbor, and selects the smaller value. The keepalive timer is then set based on the negotiated holdtime and the configured keepalive time.

Examples

```
switch(config-bgp) # timers bgp 100 150  
switch(config-bgp) # no timers bgp
```

vrf

Syntax

`vrf <VRF-NAME>`

`no vrf <VRF-NAME>`

Description

This command configures a named VRF instance `<VRF-NAME>`. After creating the VRF instance, the command prompt enters into the VRF configure context.

The `no` form of the command deletes the VRF instance from the router.

Command context

`config`

Parameters

<VRF-NAME>

Required: Specifies the VRF name. Length: Up to 32 alphanumeric characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring VRF for a named customer:

```
switch(config)# vrf cust_A
```

Removing the VRF configuration with the name **test**:

```
switch(config)# no vrf test
```

Overview

Route policies control routing paths by filtering and modifying routing information using route maps. Routing policies can filter advertised, received, and redistributed routes. They can also modify attributes for specific routes.

Configuring a route policy is two-step process:

1. Configure filters based on route attributes, such as destination address and the advertising router address.
2. Create a route map and apply filters to it.

Route maps

Route maps can be used for route redistribution. Route map entries consist of a list of match and set criteria. Match criteria specify match conditions for incoming routes. Set criteria specify the action taken if match criteria are met.

Users can configure multiple entries in the same route map. These entries contain the same route map name and are differentiated by a sequence number. A sequence number will automatically be assigned if user does not provide one.

Create a route map with one or more route map entries arranged by the sequence number under a unique route map name. The route map entry has the following parameters:

- Sequence number
- Permission (permit or deny)
- Match criteria
- Set changes

By default, a route map processes routes starting from the lowest sequence number to the highest. Other filtering mechanisms include Prefix List, AS Path List, and Community List.

Match criteria

You can use a variety of criteria to match a route in a route map. Some criteria, such as BGP community lists and AS-Path Lists, are applicable only to a specific routing protocol. Other criteria, such as the Prefix lists, can be used for any route.

When a routing protocol processes a route through a route map, it compares the route to each of the configured match statements. If the route or packet matches the configured criteria, the routing protocol processes it based on the permit or deny configuration for that match entry in the route map.

The match categories and parameters are as follows:

- **BGP parameters:** Match based on AS-path Lists, Community Lists, or Origin code.
- **Prefix lists:** Match based on an address or range of addresses.
- **Other parameters:** Match based on an IP next-hop address or metric.

Set changes

Once a route matches an entry in a route map, the route can be changed based on one or more configured set statements.

Set changes are as follows:

- **BGP parameters:** Change the AS-path, community, local preference, origin, or weight attributes.
- **Metrics:** Change the route metric.
- **Other parameters:** Change the forwarding address or the IP next-hop address.

IP prefix lists

Use prefix lists to permit or deny an address or range of addresses. Filtering by a prefix list involves matching the prefixes of routes with the prefixes listed in the prefix list. An implicit deny is assumed if a prefix does not match any entries in a prefix list. Prefix lists can be configured for both IPv4 and IPv6 addresses.

Users can configure multiple entries in a prefix list and permit or deny the prefixes that match the entry. Each entry has an associated sequence number.

AS-path lists for BGP

Use AS-path lists to filter inbound or outbound BGP route updates. If the route update contains an AS-path attribute that matches an entry in the AS-path list, the router processes the route based on the configured permit or deny condition.

Configure multiple AS-path entries in an AS-path list by using the same AS-path list name. The router processes the first matching entry. You can also configure an AS-path list to compare the AS numbers against a regular expression.

Community lists for BGP

Filter BGP route updates based on the BGP community attribute by using community lists in a route map. Match the community attribute based on a community list and set the community attribute using a route map.

A community list contains one or more community attributes. If a user configures more than one community attribute in the same community list entry, the BGP route must match all listed community attributes to be considered a match.

Configure multiple community attributes as individual entries in the community list by using the same community list name. The router processes the first community attribute that matches the BGP route, using the permit or deny configuration for that entry.

Configure community attributes in the community list in one of the following formats:

- A named community attribute, such as Internet or no-export.
- In aa:nn format.

Route flap dampening

To maintain scalability of a routed internet, it is necessary to reduce the amount of change in routing state propagated by BGP in order to limit processing requirements. The primary contributors of processing load resulting from BGP updates are the BGP decision process and adding and removing forwarding entries.

A route flap occurs when the state of a route changes from up to down or down to up. When a route state changes, the route tables of the devices that support the route also change. Route flap dampening prevents BGP from selecting unstable routes. If an EBGP peer goes down after you configure this feature, routes coming from the peer are dampened but not deleted.



NOTE: This feature applies to EBGP routes but not IBGP routes.

Route redistribution and route maps

Use route maps to control the redistribution of routes between routing domains. Route maps match on the attributes of the routes to redistribute routes that pass the match criteria. The route map can also modify the route attributes using set changes.

The router matches redistributed routes against each route map entry.

- If there are multiple match statements, the route must pass all match criteria.
- If a route passes the match criteria defined in a route map entry, actions defined in the entry are executed.
- If the route does not match the criteria, the router compares the route against subsequent route map entries.

Route processing continues until a match is made or the route is processed by all entries in the route map with no match. If the router processes the route against all entries in a route map with no match, the router accepts the route (inbound route maps) or forwards it (outbound route maps).

Route policy and route map commands

General or filtering commands

`ip aspath-list`

Syntax

```
ip aspath-list <ASPATH-LIST-NAME> [seq <SEQ>] {permit | deny} <REGEXP>
```

```
no ip aspath-list <ASPATH-LIST-NAME> [seq <SEQ>] {permit | deny} <REGEXP>
```

Description

Configures an AS Path list to match a specific AS path. AS Path lists are named lists of regular expression rules. They are used to match AS Path attributes in the routes for inclusion in or exclusion from route

policies. The sequence number is optional and is autogenerated whenever it is not explicitly mentioned. All AS Path list rules with the same name are grouped together.

The `no` form of this command removes the AS Path list configuration.

Command context

`config`

Parameters

<ASPATH-LIST-NAME>

Specifies the name of the AS Path list.

seq <SEQ>

Specifies the order of reference of the regular expression rules.

{permit | deny}

Specifies whether the route is available for further processing when there is a match.

<REGEXP>

Specifies the regular expression to match the AS Path. Standard regular expression wildcards are supported. The `_` character can be used to match the AS Path boundary.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring an AS Path list with sequence numbering:

```
switch(config)# ip aspath-list ASLst seq 5 permit _4*
```

Configuring a prefix list without sequence numbering:

```
switch(config)# ip aspath-list ASLst permit _4*
```

ip community-list

Syntax

```
ip community-list standard <COMM-LIST-NAME> <DESCRIPTION> [seq <SEQ>] {permit | deny} <COMMUNITY-NUMBER>
```

```
no ip community-list standard <COMM-LIST-NAME> <DESCRIPTION> [seq <SEQ>] {permit | deny} <COMMUNITY-NUMBER>
```

Description

Configures a community list to match a specific community number attribute. Community-list is a named list of regular expressions. They are used to match the community number attributes in the routes for inclusion in, or exclusion from route policies. The sequence number is optional and is autogenerated whenever it is not explicitly mentioned. All community-list rules with the same name are grouped.

The `no` form of this command removes the community list configuration.

Command context

`config`

Parameters

<COMM-LIST-NAME>

Specifies the name of the community list that matches community number of a route.

<DESCRIPTION>

Specifies the description of the community list. Maximum character limit is 80.

seq <SEQ>

Specifies the order of reference of the regular expression rules.

{permit | deny}

Specifies whether the route is available for further processing when there is a match.

<COMMUNITY-NUMBER>

Specifies the community number. The community number must be in AA: NN format or from the list of well-known community.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring a community list with sequence numbering:

```
switch(config)# ip community-list standard CommLst seq 5 permit 101:41
```

Configuring a community list without sequence numbering:

```
switch(config)# ip community-list standard CommLst no-export permit
```

ip prefix-list

Syntax

```
ip prefix-list <PREFIX-LIST-NAME> [seq <SEQ>] {ip | ipv6} <IP-PREFIX/MASK> [ge <0-128>] [le <0-128>]
```

```
no ip prefix-list <PREFIX-LIST-NAME> [seq <SEQ>] {ip | ipv6} <IP-PREFIX/MASK> [ge <0-128>] [le <0-128>]
```

Description

Configures a prefix list to match a set of prefixes. Prefix lists are named lists of route prefixes. They are used to match routes for inclusion in or exclusion from route policies. The sequence number determines the order of matching. The matches are performed starting from the lowest sequence number to the highest sequence number until there is a match. The sequence number is however optional and is autogenerated whenever it is not explicitly mentioned. All prefixes with the same prefix list name are grouped.

The autogenerated sequence number is derived by adding 10 to the highest sequence number available. This technique makes it possible to insert new prefix list sequence number in between.

The `ge` and `le` parameters are used to combine prefixes with a range of network mask. For example, `172.131.0.0/16 ge 16 le 24` will match all prefixes within the `172.131.0.0/16` network that have a mask greater than or equal to 16 bits and less than or equal to 24 bits in length. For instance, `172.131.1.0/18` would match, because its length is between 16 and 24 but `172.0.0.0/8` or `172.131.1.128/25` would not match.

The `no` form of this command removes the prefix list configuration.

Command context

`config`

Parameters

<PREFIX-LIST-NAME>

Specifies the name of the prefix list.

seq <SEQ>

Specifies the order of reference of the prefix rules.

{permit | deny}

Specifies whether the route is available for further processing when there is a match.

<IP-PREFIX/MASK>

Specifies the IP prefix or mask.

ge <0-32>

Specifies the minimum prefix length to be matched.

le <0-32>

Specifies the maximum prefix length to be matched.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring a prefix list with sequence numbering:

```
switch(config)# ip prefix-list PFXLST seq 5 permit 4.0.0.0/8 ge 9 le 12
```

Configuring a prefix list without sequence numbering:

```
switch(config)# ip prefix-list PSXLST permit 5.0.0.0/8
```

route-map

Syntax

```
route-map <NAME> {permit | deny} seq <NUMBER>
```

```
no route-map <NAME> {permit | deny} seq <NUMBER>
```

Description

Configures a route map entry with the given name and action by taking the CLI in the route map context. All route map entries with the same name belong to the same route map. The route map entry rules are processed in order by sequence number, until a match is found.

The `no` form of this command removes the route map entry configuration.

Command context

config

Parameters

<NAME>

Specifies the name of the route map. Required.

{permit | deny}

Specifies whether the route is available for further processing when there is a match. Required.

<NUMBER>

Specifies the sequence number of the entry. Required.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring a route map entry:

```
switch(config)# route-map GlobalMap permit seq 10
```

Removing a route map entry configuration:

```
switch(config)# no route-map GlobalMap permit seq 10
```

continue

Syntax

```
continue <SEQUENCE NUMBER>
```

```
no continue
```

Description

Allows you to execute additional entries in a route map. The sequence number specifies the route-map entry's sequence number that will be executed next if the existing entry's match clause is successful.

If a successful match occurs and continue command exists, the route map saves the set value first and then jumps to the specified route map entry.

Set clauses are saved during the match clause evaluation and are executed only after the route map evaluation is completed. The set clauses are executed in the order in which they were configured.

Set clauses can be accumulative or additive as `set as-path prepend` or it can be absolute as `set metric`. For set commands that configures an accumulative value, subsequent values are added in order in which they were configured. For set commands that configures an absolute value, The values from the last instance will be applied.

The `no` form of this command removes the route map continue configuration.



NOTE: If the specified route-map sequence entry does not exist, route-map processing will be terminated at the current sequence number if its clause is matched. The continue sequence number must be higher than the current route map sequence number for this command to take effect.

Command context

```
config
```

Parameters

<SEQUENCE NUMBER>

Specifies the value of the route map entry to be executed next after a successful match clause.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring a route map to continue to execute an additional entry:

```
switch(config)# route-map GlobalMap permit 10  
switch(config-route-map-GlobalMap-10)# continue 40
```

Removing a route map continue configuration:

```
switch(config)# route-map GlobalMap permit seq 10  
switch(config-route-map-GlobalMap-10)# no continue 40
```

Match commands

match aspath-list

Syntax

```
match aspath-list <ASPATH-LIST-NAME>  
  
no match aspath-list <ASPATH-LIST-NAME>
```

Description

Matches the AS path attribute of the route with one or more regular expressions in the AS path list.

The **no** form of this command restores the default behavior of not matching the AS path attribute of the route.

Command context

config-route-map

Parameters

<ASPATH-LIST-NAME>

Specifies the name of the AS path list to match the AS path attribute of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the AS path list:

```
switch(config)# ip aspath-list ASLst permit 1001  
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# match aspath-list ASLst
```

match community-list

Syntax

```
match community-list <COMMUNITY-LIST-NAME> [exact-match]  
  
no match community-list <COMMUNITY-LIST-NAME> [exact-match]
```

Description

Matches the community number attribute of the route with one, or more regular expressions in the community-list.

The `no` form of this command restores the default behavior of not matching the community number attribute of the route.

Command context

`config-route-map`

Parameters

<COMMUNITY-LIST-NAME>

Specifies the name of the community-list to match the community number attribute of the route.

[exact-match]

Indicates that the community number attribute must match exactly with the expressions in the community-list. However, the order of the communities in the community-list is of no significance.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the community list:

```
switch(config)# ip community-list standard CommLst 101:41 permit 12:201
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match community-list CommLst
```

match interface

Syntax

`match interface <INTERFACE-NAME>`

`no match interface <INTERFACE-NAME>`

Description

Matches the outgoing interface value of the route with the value configured in the match clause.

The `no` form of this command restores the default behavior of not matching the outgoing interface value of the route.

Command context

`config-route-map`

Parameters

<INTERFACE-NAME>

Specifies the value to be matched with the outgoing interface of the route entry.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the outgoing interface of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match interface 1/1/1
```

match ip address prefix-list

Syntax

```
match ip address prefix-list <PREFIX-LIST-NAME>

no match ip address prefix-list <PREFIX-LIST-NAME>
```

Description

Matches the destination IP address prefix of the routes with one or more addresses in the prefix list.

The `no` form of this command restores the default behavior of not matching the destination IP address prefix of the routes to their default value.

Command context

config-route-map

Parameters

<PREFIX-LIST-NAME>

Specifies the name of the prefix list to be matched with the network address of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a prefix list and a match clause in route map to match the prefix list:

```
switch(config)# ip prefix-list PfxLst permit 4.0.0.0/8
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match ip address prefix-list PfxLst
```



NOTE: When the IP prefix list with prefix and mask-length of `0.0.0.0/0` is used, the route matches default-route `0.0.0.0/0` as well as any other route. This behavior would be changed to match only the default-route in the next release.

match ip next-hop

Syntax

```
match {ip | ipv6} next-hop {<ADDRESS> | prefix-list <PREFIX-LIST-NAME>}

no match {ip | ipv6} next-hop [<ADDRESS> | prefix-list <PREFIX-LIST-NAME>]
```

Description

Matches the next-hop address of the route with the configured address in the match clause.

The `no` form of this command restores the default behavior of not matching the next-hop address of the route.

Command context

config-route-map

Parameters

<ADDRESS>

Specifies the IPv4 address to match with the next-hop address of the route.

prefix-list <PREFIX-LIST-NAME>

Specifies the name of the IP prefix list to be matched with the next-hop address of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the next-hop address of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match ip next-hop 1.1.1.2
```

match ip route-source

Syntax

match ip route-source prefix-list <PREFIX-LIST-NAME>

no match ip route-source prefix-list <PREFIX-LIST-NAME>

Description

Matches the IP address of the source of the route using IP prefix lists.

The **no** form of this command restores the default behavior of not matching the IP address of the route.

Command context

config-route-map

Parameters

<PREFIX-LIST-NAME>

Specifies the name of the prefix list to match the IP address of the source of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the source of the route:

```
switch(config)# ip prefix-list RouterLst 4.4.4.4/32
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match ip route-source prefix-list RouterLst
```

match local-preference

Syntax

```
match local-preference <VALUE>
```

```
no match local-preference <VALUE>
```

Description

Matches the local preference value of the route with the value configured in the match clause.

The `no` form of this command restores the default behavior of not matching the local preference value of the route.

Command context

```
config-route-map
```

Parameters

<VALUE>

Specifies the value to be matched with the route entry local preference in the range of 1 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the local preference of the route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# match local-preference 100
```

match metric

Syntax

```
match metric <VALUE>
```

```
no match metric <VALUE>
```

Description

Matches the MED value of the route with the value configured in the match clause.

The `no` form of this command restores the default behavior of not matching the MED value of the route.

Command context

```
config-route-map
```

Parameters

<VALUE>

Specifies the value to be matched with the route entry MED.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the metric of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match metric 10
```

match origin

Syntax

```
match origin {igp | egp | incomplete}

no match origin [igp | egp | incomplete]
```

Description

Matches the route origin attribute of the route with route configured in the match clause.

The `no` form of this command restores the default behavior of not matching the route origin attribute of the route.

Command context

config-route-map

Parameters

{igp | egp | incomplete}

Specifies if the route origin attribute is matched with a match clause which originated as IGP, EGP, or has unknown origin. The unknown origin is typically redistributed from another routing protocol.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the origin:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# match origin igp
```

match route-type

Syntax

```
match route-type external {type-1 | type-2}

no match route-type external [type-1 | type-2]
```

Description

Matches the metric-type value of the OSPF external route with the value configured in the match clause.

The `no` form of this command restores the default behavior of not matching the metric-type value of the OSPF external route.

Command context

config-route-map

Parameters

{**type-1** | **type-2**}

Specifies the *type-1* or *type-2* OSPF value to be matched with the external route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the metric-type value of the OSPF external route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# match route-type external type-1
```

match source-protocol

Syntax

```
match source-protocol {bgp | connected | ospf | static}
```

```
no match source-protocol [bgp | connected | ospf | static]
```

Description

Matches the source routing protocol value of the route with the value configured in the match clause.

The `no` form of this command restores the default behavior of not matching the source routing protocol value of the route.

Command context

config-route-map

Parameters

{**bgp** | **connected** | **ospf** | **static**}

Specifies the *bgp*, *connected*, *ospf*, or *static* value to be matched with the route entry source protocol.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in the route map to match the source protocol route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# match source-protocol ospf
```

match tag

Syntax

```
match tag <value>
```

```
no match tag <value>
```

Description

Matches the tag value of the route with the one configured in the match clause. Applies to static routes that will be redistributed to ospfv2 and ospfv3 protocols.

The `no` form of this command removes the tag value of the route.

Command context

config-route-map

Parameters

value

Numeric value to match with the route tag. Required.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a match clause in route-map to match the tag value of the route:

```
switch(config)# route-map GlobalMap permit 11
switch(config-route-map)# match tag 20
```

Set commands

set as-path exclude

Syntax

```
set as-path exclude <AS>
```

```
no set as-path exclude <AS>
```

Description

Removes all occurrences of the configured AS Path from the AS Path attribute of the route.

The `no` form of this command restores the default behavior of not modifying the AS Path attribute list.

Command context

config-route-map

Parameters

<AS>

Specifies the AS number to be removed from the AS Path attribute of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to remove the AS from the AS Path attribute of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# set as-path exclude 1001
```


set as-path prepend

Syntax

```
set as-path prepend <AS> <AS>...  
  
no set as-path prepend <AS> <AS>...
```

Description

Prepends the list of the configured AS numbers to the AS Path attribute of the routes. To ensure that the AS path conforms to standards, the local AS is prepended after this command is executed.

The `no` form of this command restores the default behavior of not modifying the AS Path attribute list.

Command context

config-route-map

Parameters

<AS> <AS>...

Specifies the AS numbers to be prepended from the AS Path attribute of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to prepend the AS from the AS Path attribute of the route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# set as-path prepend 1  
switch(config-route-map-GlobalMap-11)# no set as-path prepend 102
```



NOTE: The `no` form of the command deletes the entire list of AS-Path prepend configuration regardless of the parameter list. In this example, the `no` form command would result in deletion of all the three AS numbers that were earlier configured.

set community

Syntax

```
set community {<AA:NN> | internet | no-export | no-advertise | local-as} [additive | delete]  
  
no set community [<AA:NN> | internet | no-export | no-advertise | local-as] [additive | delete]
```

Description

Modifies the community number attribute of the route with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the community number attribute of the route.

Command context

config-route-map

Parameters

{<AA:NN> | **internet** | **no-export** | **no-advertise** | **local-as**}

Selects the value to be set as the community number attribute of the route in the AA: nn format or as a known community name **internet**, **no-export**, **no-advertise**, and **local-as**.

[**additive**]

Specifies that the specified community number is added to the existing community number attribute of the route.

[**delete**]

Specifies that the specified community number is removed from the existing community number attribute of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the community number attribute of the route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# no set community 11:101  
switch(config-route-map-GlobalMap-11)# set community no-advertise
```

set dampening

Syntax

```
set dampening {half-life <VALUE> reuse <VALUE> suppress <VALUE> max-suppress-time <VALUE>}
```

```
no set dampening
```

Description

Sets parameters of route flap dampening feature.

Parameters

half-life

Time to reduce the penalty to half.

reuse

Lower threshold of penalty.

suppress

Upper threshold of penalty.

max-suppress-time

Max time to keep route suppressed.

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch(config-route-map-abc-20)# set dampening half-life 5 reuse 50 suppress 125 max-suppress-time 255  
switch(config-route-map-abc-20)# no set dampening half-life 5 reuse 50 suppress 125 max-suppress-time 255
```

```
switch(config-route-map-abc-20) # set dampening  
switch(config-route-map-abc-20) # no set dampening
```

set ip nexthop

Syntax

```
set {ip | ipv6} nexthop {global} <IP-ADDR>  
  
no set {ip | ipv6} nexthop {global} <IP-ADDR>
```

Description

Sets the IP address of the next-hop of the route with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the IP address of the next-hop of the route.

Command context

config-route-map

Parameters

<IP-ADDR>

Specifies the IPv4 address to be set as the next-hop address of the route.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the next-hop address of the route entry:

```
switch(config) # route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11) # set ip nexthop 1.1.1.2
```

set local-preference

Syntax

```
set local-preference <VALUE>  
  
no set local-preference <VALUE>
```

Description

Modifies the local-preference attribute of the route entry with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the local-preference attribute of the route entry.

Command context

config-route-map

Parameters

<VALUE>

Specifies the value to be set as the local-preference attribute of the route entry. Range: 0 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the metric value of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# set local-preference 100
```

set metric

Syntax

```
set metric <VALUE>
```

```
no set metric <VALUE>
```

Description

Modifies the metric value of the route with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the metric value of the route.

Command context

config-route-map

Parameters

<VALUE>

Specifies the value to be set as the metric value of the route. Range: 0 to 4294967295.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the metric value of the route:

```
switch(config)# route-map GlobalMap permit seq 11
switch(config-route-map-GlobalMap-11)# set metric 10
```

set origin

Syntax

```
set origin [igp | egp | incomplete]
```

```
no set origin [igp | egp | incomplete]
```

Description

Modifies the route origin attribute of the route update with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the route origin attribute of the route.

Command context

config-route-map

Parameters

{**igp** | **egp** | **incomplete**}

Selects the route update originated to IGP, EGP, or incomplete. When incomplete is selected, the route update origin is set to unknown.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the origin attribute of the route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap-11)# set origin igp
```

set tag

Syntax

set tag <value>

[no] set tag <value>

Description

Modifies the tag value of the route with the one configured in the set clause. Applicable to static routes that will be redistributed to ospfv2 and ospfv3 protocols.

The **no** form of this command removes the set clause tag value.

Command context

config-route-map

Parameters

value

Numeric value to change the route entry tag. Range: 0-4294967295. Required.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in route-map to modify the tag value of the route:

```
switch(config)# route-map GlobalMap permit 11  
switch(config-route-map)# set tag 10
```

set weight

Syntax

```
set weight <VALUE>
```

```
no set weight <VALUE>
```

Description

Modifies the weight attribute of the route with the value configured in the set clause.

The `no` form of this command restores the default behavior of not modifying the weight attribute of the route.

Command context

config-route-map

Parameters

<VALUE>

Specifies the value to be set as the weight attribute of the route. Range: 0 to 65535.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a set clause in the route map to modify the metric value of the route:

```
switch(config)# route-map GlobalMap permit seq 11  
switch(config-route-map-GlobalMap=11)# set weight 100
```

Show commands

show ip aspath-list

Syntax

```
show ip aspath-list [<NAME>] [vsx-peer]
```

Description

Shows the configuration details of the AS path list.

Command context

Manager (#)

Parameters

<NAME>

Specifies name of the AS path list.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

Showing the IP AS path list configuration information:

```
switch# show ip aspath-list
ip aspath-list ASLst
  seq 10 permit 22 33
  seq 20 deny 44
```

show ip community-list

Syntax

```
show ip community-list [<NAME>] [vsx-peer]
```

Description

Shows the configuration details of the community-list.

Command context

Manager (#)

Parameters

<NAME>

Specifies name of the community-list.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

Showing the community list configuration information:

```
switch# show ip community-list
ip community-list standard CommLst
  seq 10 permit 11:101
  seq 20 deny 12:201
```

show ip prefix-list

Syntax

```
show ip prefix-list [<NAME>] [vsx-peer]
```

Description

Shows the configuration details of the IP prefix lists.

Command context

Manager (#)

Parameters

<NAME>

Specifies name of the IP prefix lists.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

Showing the IP prefix list configuration information:

```
switch# show ip prefix-list
ip prefix-list PfxLst: 2 entries
  seq 10 permit 3.0.0.0/8 ge 8 le 8
  seq 20 deny 4.0.0.0/8 ge 8 le 8
```

```
switch# show ipv6 prefix-list
ipv6 prefix-list x: 1 entries
  seq 10 permit 2011::/64 ge 64 le 64
```

show route-map

Syntax

```
show route-map [<NAME>] [vsx-peer]
```

Description

Shows the configuration details of the route map.

Command context

Manager (#)

Parameters

<NAME>

Specifies name of the route map.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

Showing the route map configuration information:

```
switch# show route-map
Route map: InternetFilter
  Seq 10, permit,
    Match :
      origin : egp
      metric : 123
    Set :
      community : 23:34
      metric : 3
      as_path_exclude : 123
      local_preference : 3456
      origin : igp
      weight : 25
  Seq 20, permit,
    Match :
      origin : egp
      metric : 456
    Set :
      community : 44:44
      metric : 5
      as_path_prepend : 444
      local_preference : 66
      origin : igp
      weight : 250

Route map: LocalFilter
  Seq 10, permit,
    origin : egp
    metric : 10
    ip next-hop address : 2.2.2.3
    local-preference : 20
    route-type : external_type1
    source-protocol : static
    prefix-list : PfxLst
    aspath-list : ASLst
    community-list : CommLst
    ip next-hop prefix-list : PfxLst
    ip route-source prefix-list : PfxLst
  Set :
    community : 22:33
    metric : 25
    as_path prepend : 65535 65534
    ip next-hop address : 2.2.2.4
    local_preference : 30
    origin : igp
    weight : 30
    dampening : half-life = 5, reuse = 50, suppress = 125, max-suppress-time = 15
```

All the match clauses are grouped. All the set clauses are grouped.

Overview

Equal Cost Multipath (ECMP) allows a router to load balance traffic destined to some network that is reachable through multiple equal cost route nexthops. ECMP functionality is dependent on L3 routing being enabled on the router.

The nexthop chosen out of the ECMP group of nexthops for a given packet is typically determined by its header information, usually by hashing source and destination IP addresses as well as potentially source and destination L4 ports. The hashing algorithm used is dependent on the type of switch (in this case the Aruba 8400) in use as well as whether the packet is flowing through the slow-path or fast-path.

By default all of the following fields are enabled and utilized for the load balancing decision. Configuration to exclude fields is not available on the Aruba 8400.

- L3 Destination IP address
- L3 Source IP address
- L4 Destination port
- L4 Source port

ECMP support is NOT dependent on address family. The ECMP feature applies to both IPv4 and IPv6 traffic flowing through the router.



NOTE: If two different routing protocols are (mis)configured to use the same Administrative Distance, then they have the opportunity to create ECMP groups together.

ECMP commands

show ip ecmp

Syntax

```
show ip ecmp [vsx-peer]
```

Description

Displays the Equal Cost Multipath (ECMP) configuration.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Usage

On the Aruba 8400, the following fields are enabled by default and cannot be excluded: Source IP, Destination IP, Source Port, Destination Port.

Examples

```
switch# show ip ecmp

ECMP Configuration
-----

ECMP Status          : Enabled

ECMP Load Balancing by
-----
Source IP            : Enabled
Destination IP       : Enabled
Source Port          : Enabled
Destination Port     : Enabled
```

Overview

This section describes how to configure and verify VRRP configuration and operational states.

- All VRRP configurations work in VRRP context.
- VRRP can be configured on physical ports, VLAN interfaces, and LAG interfaces.
- VRRP mandates the associated interface to be routing interface.

In many networks, edge devices are configured to send packets to a statically configured default router. If this router becomes unavailable, devices that use it as a first-hop router become isolated from the network.

VRRP uses dynamic failover to ensure the availability of an end node default router. The IP address used as the default route is assigned to a virtual router (VR). The VR includes:

- A Master router assigned to forward traffic designated for the virtual route
- One or more prioritized Backup routers (If a Backup is forwarding traffic for the VR, it has replaced the Owner as the Master router.)

This redundancy provides a backup for gateway IP addresses (first-hop routers). If a VR Master router becomes unavailable, the traffic it supports will be transferred to a Backup router without major delays or operator intervention. This operation can eliminate single-point-of-failure problems and provide dynamic failover (and failback) support. As long as one physical router in a VR configuration is available, IP addresses assigned to the VR are always available. Edge devices can send packets to these IP addresses without interruption.

Advantages to using VRRP include:

- Minimizing failover time and bandwidth overhead if a primary router becomes unavailable
- Minimizing service disruptions during a failover
- Providing backup for a load-balanced routing solution
- Avoiding the need to make configuration changes in the end nodes if a gateway router fails
- Eliminating the need for router discovery protocols to support failover operation

Terminology

Owner router: The Owner router is automatically configured with the highest VRRP router priority in the VR (255). It operates as the Master router for the VR unless it becomes unavailable to the network. All Virtual Router members can be configured so virtual IP is not the same as physical (or real) IP. Such virtual address can be called pure virtual IP address.

Master router: The Owner or Backup router that is the physical forwarding agent for routed traffic using the VR as a gateway. There can be only one router operating as the Master for a network. If the router configured as the Owner for a VR is available to the network, it will also be the Master. If the Owner fails or loses availability to the network, the highest-priority Backup becomes the Master.

Backup: A router configured in a VR as a Backup to the Owner/Master for the same VR. There must be a minimum of one Backup in a VR to support VRRP operation if the Owner/Master fails. Every backup is created with a configurable priority (default: 100). This priority determines the precedence for becoming the Master of the VR if the Owner or another Backup operating as Master becomes unavailable.

When the current Master router is no longer available, the Backup router with highest priority will become current master. When a router with higher priority becomes available, it is switched to master. The user can override this behavior by disabling preemption mode.

Virtual Router (VR): Consists of one Owner/Master router and one or more Backup routers that belong to the same network. The Owner is the router that owns any IP addresses associated with the VR. The VR has one virtual IP address (multiple virtual IP addresses for multinetted interfaces) that correspond to a real IP address on the Owner. A VR includes the following:

- A VR identification (VRID) configured on all VRRP routers in the same network. For a multinetted interface, the VRID is configured on all routers in the same subnet.
- The same VIP configured on each instance of the same VR.
- The status of either Owner or Backup configured on each instance of the same VR. There can be one Owner and one or more Backups configured on each VR.
- The priority level configured on each instance of the VR. On the Owner router, the highest priority setting of 255 is automatically fixed. On Backups, the default priority setting is 100 and configurable.
- A VR MAC address that is not configurable.

Virtual IP address (VIP): The VIP associated with a VR must be a real IP address already configured in the associated interface on the Owner router.

- If the VIP is an IPv6 address, a link-local address must be configured before adding a global IPv6 address.
- The Owner and all Backup routers belonging to the VR have this IP address configured in their VRID contexts as the VIP.
- A subnetted interface allows multiple VIPs. If there are fewer IP addresses in an interface and a user wants VRRP support on multiple subnets, configure a separate VR instance for each IP address in the interface.

VRID: The identifier for a VR configured on an interface. A VRID can be used for only one VR in any interface on a router. It can be used again for a different VR in a different interface.

VRRP operation

VRRP supports router redundancy through a prioritized election process among routers configured as members of the same virtual router (VR).

A VR includes two or more member routers configured with:

- A virtual IP address that is also configured as a real IP address on one of the routers
- A virtual router MAC address

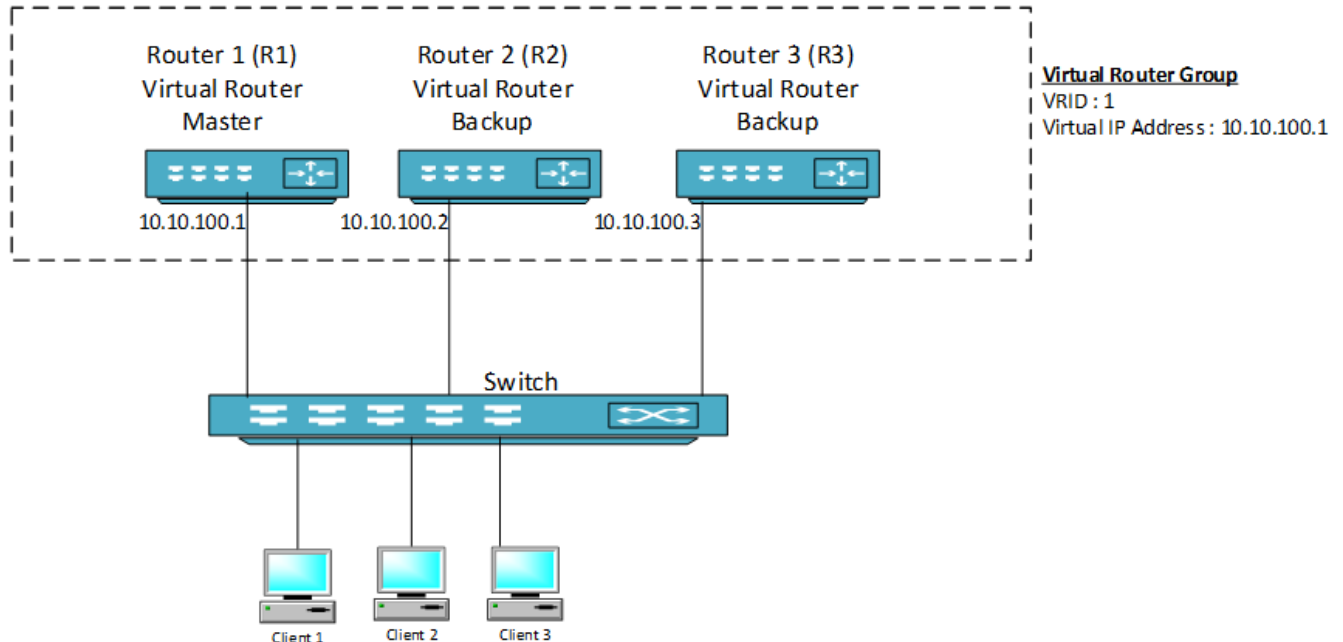
The router that owns the IP address is configured to operate as the Owner of the VR for traffic-forwarding purposes. By default, this router has the highest VRRP priority in the VR. Any other routers in the VR have a lower priority and are configured to operate as Backups if the Owner router becomes unavailable.

The Owner normally operates as the Master for a VR. If the Owner becomes unavailable, a failover to a Backup router belonging to the same VR occurs. This Backup becomes the Master. If the Owner recovers, a failback occurs, and Master status reverts to the Owner.

**NOTE:**

- Using more than one Backup provides additional redundancy. If the Owner and the highest-priority Backup fail, another lower-priority Backup can take over as Master.
- In a VRRP owner scenario, DAD should be disabled for VRRP to work.

The image shown here depicts a basic VLAN topology. In this example, Routers 1, 2 and 3 form a VRRP group. The IP address of the group is the same address configured for the SVI interface of R1 (10.10.100.1).



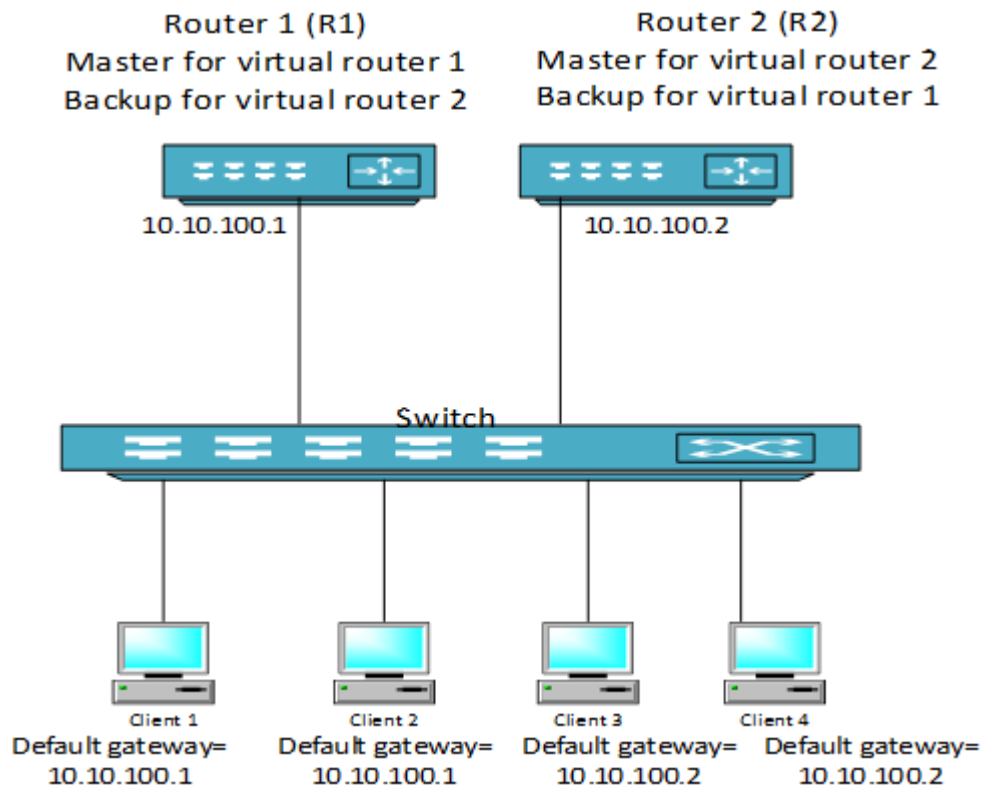
Because the virtual IP address uses the IP address of the SVI interface of Router R1, Router R1 is the master (also known as IP address Owner).

- As the master, R1 owns the virtual IP address of the VRRP group and forwards packets sent to this IP address.
- Clients 1 through 3 are configured with the default gateway IP address of 10.10.100.1.
- Routers R2 and R3 function as backups. If the master fails, the backup router with highest priority becomes the master and takes over the virtual IP address to provide uninterrupted service for the clients connected to L2 Switch. When Router R1 recovers, it becomes master again.

Multiple VRRP groups

A user can configure multiple VRRP groups on a router (or L3) interface (Physical, LAG, or SVI interfaces). In a topology in which multiple VRRP groups are configured on a router interface, the interface can act as a master for one VRRP group and a backup for one or more other VRRP groups.

This image displays a topology in which VRRP is configured so that Routers R1 and R2 share the traffic to and from clients 1 through 4. Routers R1 and R2 act as backups to each other if either router fails.



This topology contains two virtual IP addresses for two VRRP groups that overlap. For VRRP group 1, Router R1 is the owner of IP address 10.10.100.1 and is the Master. Router R2 is the Backup to Router R1. Clients 1 and 2 are configured with the default gateway IP address of 10.10.100.1.

For VRRP group 2, Router R2 is the owner of IP address 10.10.100.2 and is the Master. Router R1 is the Backup to Router R2. Clients 3 and 4 are configured with the default gateway IP address of 10.0.0.2.

VRRP priority

In a Backup router VR configuration, the virtual router priority defaults to 100. The priority for the configured Owner is automatically set to the highest value of 255.

In a VR in which there are two or more Backup routers, priority settings can be reconfigured to define the order in which Backups will be reassigned as Master if a failover occurs.

VRRP preemption

In a Backup router VR configuration, the virtual router priority defaults to 100. The priority for the configured Owner is automatically set to the highest value of 255.

When multiple Backup routers exist in a VR:

- If the current Master fails and the highest-priority Backup is not available, VRRP selects the next-highest priority Backup to operate as Master.
- If the highest-priority Backup later becomes available, it preempts the lower-priority Backup and takes over the Master function.

To remove this preemptive ability on a VR, disable it with the `no preempt` command.



NOTE: Preemption applies only to VRRP routers configured as Backups.

Virtual Router MAC address

When a Virtual Router (VR) instance is configured, the protocol automatically assigns a MAC address based on the standard MAC prefix for VRRP packets plus the VRID number (as described in RFC 3768). The first five octets form the standard MAC prefix for VRRP and the last octet is the configured VRID. For example:

```
00-00-5E-00-01-< VRid >
```

VRRP and ARP

The current Master for a VR responds to ARP requests for the virtual IP addresses with the VR-assigned MAC address. The virtual MAC address is also used as source MAC address for periodic advertisements sent by the current Master.

The VRRP router responds to ARP requests for non-virtual IP addresses with the system MAC address. Non-virtual IP addresses are not configured as virtual IP addresses for any VR on the interface.

VRRP and MLAG

Users can enable VRRP on SVI interfaces (or Layer 3 VLANs) that have multi-chassis lag (MLAG) ports as member ports. MLAG allows links that are physically connected to two different Aruba 8400 devices to appear as a single lag port.

A router performs a Layer 3 route table lookup and Layer 3 forwarding when the destination MAC in the Ethernet frame matches its own MAC address (or virtual MAC address when the router is acting as a VRRP master). Otherwise the packets are switched.

In a topology with MLAG and VRRP enabled, an MLAG switch acting as a VRRP backup could receive IP packets with the virtual MAC address as the destination MAC. In this scenario, the MLAG switch acting as a VRRP backup switches the traffic to peer (or VRRP master) using inter-switch-link and peer (VRRP master) performs the Layer 3 forwarding.

VRRP tracking

VRRP supports object tracking. This functionality allows monitoring of the state of a configured object. The state determines the priority of the VRRP router in a VRRP group. A track object can be created using the `track <object-id>` command and the object can be associated with an interface. A change in interface state will then affect the priority of a VRRP group.

High availability

VRRP supports high availability through stateful restarts and stateful switchovers.

- A stateful restart occurs when the VRRP process (or daemon) fails and is restarted.
- A stateful switchover occurs when the active management module (AMM) switches to the standby management module (SMM).

VRRP and Neighbor Discovery for IPv6

Neighbor Discovery (ND) is the IPv6 equivalent of the IPv4 ARP for layer 2 address resolution. It uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of neighbors
- Verify that a neighbor is reachable
- Track neighbor (local) routers

Neighbor Discovery enables functions such as:

- Router and neighbor solicitations and discovery
- Detecting address changes for devices
- Identifying a replacement for a router or router path that has become unavailable
- Duplicate address detection (DAD)
- Router Advertisement processing
- Neighbor reachability
- Resolution of destination addresses
- Changes to link-layer addresses

An instance of Neighbor Discovery is triggered on a device when a new or changed IPv6 address is detected. VRRPv3 provides a faster failover to a backup router by not using standard ND procedures. A failover to a backup router can occur in approximately three seconds without any interaction with hosts and a minimum of VRRPv3 traffic.

Duplicate address detection (DAD)

Duplicate Address Detection verifies that a configured unicast IPv6 address is unique before it is assigned to an interface. When the Owner router fails, the backup VRRP router assumes the Master role. When the Owner router becomes operational, DAD will fail because a backup VRRP router is in the Master role that responds to the DAD request. To avoid this, on virtual routers that are in owner mode (priority = 255) DAD needs to be disabled for the interface on which the owner VR is configured.

Guidelines and limitations

- VRRP must be enabled at the global level using the `router vrrp enable` command. It must also be enabled at the interface level at which you configure VRRP.
- A maximum of 8 IPv4 and 8 IPv6 VRRP groups are supported on an interface.
- A maximum of 256 VRRP groups is supported on a router. The groups can be IPv4 or IPv6.
- Supported virtual router identification (VRID) range is 1-255. A maximum of 8 VRIDs can be used to achieve a scale of 256 VRRP groups by using the same VRID across different interfaces.
- Proxy-ARP and Active-Gateway are not supported in conjunction with VRRP.

VRRP commands

address

Syntax

```
address <IP-ADDR> [ primary | secondary ]
```

```
no address <IP-ADDR> [ primary | secondary ]
```

Description

Configures a primary or secondary IPv4 or IPv6 address for the VRRP group. To use secondary IP addresses in a VRRP group, you must first configure a primary IP address on the same group. A maximum of 16 IP addresses per IPv4 VRRP group and 8 IPv6 addresses per IPv6 VRRP group are supported.



NOTE: Do not configure an IPv4 VRRP group using addresses from the /30, /31, and /32 subnets of the interface IP address.



NOTE: 16 Virtual IP addresses per IPv4 VR and 8 Virtual IP addresses per IPv6 VR are supported.

The total number of VIPs supported by a switch is:

- 1024 VIPs for IPv4 VRs
- 512 VIPs for IPv6 VRs

The `no` form of this command deletes a primary or secondary IPv4 or IPv6 address from the VRRP group.

Command context

config

Parameters

<IP-ADDR>

Configures the IPv4 or IPv6 address.

primary

Configures a primary address.

secondary

Configures a secondary address.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# address 10.0.0.1 primary
```

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv6
switch(config-if-vrrp)# address fe80::1 primary
```

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no address 10.0.0.1 primary

switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv6
switch(config-if-vrrp)# no address fe80::1 primary
```

authentication

Syntax

```
authentication {<text> | <md5>} {<plaintext> | < ciphertext>} <key-string>
```

```
no authentication
```

Description

This command enables authentication mode and the authentication key for VRRP groups on an interface. VRRP members or routers of the same VRRP group must use the same authentication mode and authentication key.

The no form of this command disables authentication mode and the authentication key for VRRP groups on an interface.

IPv4 VRRPv3 and IPv6 VRRPv3 do not support VRRP packet authentication. Authentication mode and key configuration take effect only in VRRPv2 (IPv4 only - RFC2338).

In VRRPv3, authentication mode and authentication key settings do not take effect because VRRP Authentication was removed from RFC5798.

VRRP provides the following authentication modes as described in RFC2338:

Simple authentication

The sender fills an authentication key into the VRRP packet and the receiver compares the received authentication key with its local authentication key.

If the two authentication keys match, the received VRRP packet is legitimate. Otherwise, the received packet is illegitimate and is discarded.



NOTE: Authentication key text is sent in the clear and can be seen in a packet trace. This makes MD5 authentication more secure than text.

MD5 authentication

The sender computes a digest for the packet that will be sent using the authentication key and MD5 algorithm, and saves the result in the VRRP packet.

The receiver performs the same operation with the authentication key and MD5 algorithm, and compares the result with the content in the authentication header. If the results match, the received VRRP packet is legitimate. Otherwise, the received packet is illegitimate and is discarded.



NOTE: Key chain support for VRRP MD5 authentication is not available yet.

Command context

```
config
```

Parameters

text

Configures authentication type as simple-text.

MD5

Configures authentication type as MD5 (message-digest).

plaintext

Configures plaintext key.

ciphertext

Configures ciphertext key.

key-string

Specifies cipher key.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enable VRRP authentication:

```
switch# configure terminal
switch(config)#interface 1/1/1
switch(config-if)#vrrp 1 address-family ipv4
switch(config-if-vrrp)#version 2
switch(config-if-vrrp)#authentication md5 plaintext testvrrpkey
```

Disable VRRP authentication:

```
switch# configure terminal
switch(config)#interface 1/1/1
switch(config-if)#vrrp 1 address-family ipv4
switch(config-if-vrrp)#version 2
switch(config-if-vrrp)#no authentication
```

Ciphertext:

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# version 2
switch(config-if-vrrp)# authentication md5 ciphertext AQBapfsvhcfZ/PIzpt5nHxUZC3jIuUvDYvv7P8hP3gp53biBAAAAOjc0a8=
```

bfd <IPV4-ADDR>

Syntax

bfd <IPV4-ADDR>

no bfd <IPV4-ADDR>

Description

Enables BFD under VRRP for the specified IP address. BFD is asynchronous and echo mode is supported.

The `no` form of this command disables BFD under VRRP for the specified IP address.

Command context

config-if-vrrp

Parameters

<IPV4-ADDR>

Specifies the address on which to enable BFD in IPv4 format (x.x.x.x), where x is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

On the 6400 Switch Series, interface identification differs.

Enabling BFD on the address 10.0.0.1 on VRRP 1:

```
switch(config)# interface 1/1/1
switch(config-if)# routing
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# bfd 10.0.0.1
```

Disabling BFD on the address 10.0.0.1 on VRRP 1:

```
switch(config)# interface 1/1/1
switch(config-if)# routing
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no bfd 10.0.0.1
```

preempt

Syntax

preempt

no preempt

Description

Enables the preempt option. The default value is enabled. In default mode, a backup router with a higher priority than another backup that is operating as master will take over the master function.

Applies to VRRP backup routers only and is used to minimize network disruption caused by unnecessary preemption of the master operation among backup routers.

The **no** form of this command disables the preempt option, thus preventing the higher-priority backup from taking over the master operation from a lower-priority backup. This command does not prevent an owner router from resuming the master function after recovering from being unavailable.

Command context

config

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# preempt
```

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no preempt
```

preempt delay minimum

Syntax

```
preempt delay minimum <DELAY-IN-SECONDS>
```

```
no preempt delay minimum <DELAY-IN-SECONDS>
```

Description

Sets the time in seconds (1-3600) that the router will wait before taking control of the virtual IP and starting to route packets.

The `no` form of this command sets the preempt delay for the VRRP group to the default preempt delay of 0 seconds.

The VRRP Preempt Delay Timer (PDT) allows admin users to configure a period of time before the VR takes control of the virtual IP address. It does not transition to the Master state until the timer period expires.

The timer value configured should be long enough to allow upper layer protocol to converge. The PDT is applied only during initialization of the router.

Command context

config

Parameters

<DELAY-IN-SECONDS>

Selects the time in seconds (1-3600). Default is 0 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# preempt delay minimum 30
```

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no preempt delay
```

priority

Syntax

```
priority <1-254>
```

```
no priority
```

Description

Sets the priority for the VRRP group.

The `no` form of this command sets the priority for the VRRP group as default priority.

- The default value for non-owner virtual routers is 100.
- The default value for owner virtual router is 255, which cannot be changed.

Command context

config

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# priority 150
```

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no priority
```

router vrrp enable

Syntax

router vrrp enable

Description

Enables VRRP protocol globally. You must globally enable the VRRP feature for VRRP virtual router.

Command context

config

Authority

Administrators or local user group members with execution rights for this command.

Example

Enabling VRRP protocol globally:

```
switch(config)# router vrrp enable
```

router vrrp disable

Syntax

router vrrp disable

Description

Disables VRRP protocol globally. Does not remove all VRRP configuration.

Command context

config

Authority

Administrators or local user group members with execution rights for this command.

Example

Disabling VRRP protocol globally:

```
switch(config)# router vrrp disable
```

no router vrrp

Syntax

```
no router vrrp
```

Description

Removes VRRP configuration. If `auto-confirm` is enabled or VRRP is not configured on any interface, this command will not ask for user confirmation.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Removing VRRP configuration:

```
switch(config)# no router vrrp  
All VRRP configuration will be deleted.  
Do you want to continue (y/n)?
```

show track

Syntax

```
show track [brief | <OBJECT-ID>]
```

Description

Shows all or specific track object information.

Command context

```
Manager (#)
```

Parameters

brief

Displays brief information about all or specific track objects

<OBJECT-ID>

Displays information about a specified track object (1-128)

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

```
switch# show track
Track 1
  interface 1/1/1
  Interface is DOWN
```

```
switch# show track brief
Track  Interface      State
1       1/1/1         Down
2       None           Down
3       1/1/2         Up
```

show track brief

Syntax

```
show track brief
```

Description

Shows brief information for all track objects.

Command context

Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

Showing brief information for all track objects:

```
switch# show track brief
Track  Interface      State
1       1/1/1         Down
2       None           Down
3       1/1/2         Up
```

show vrrp

Syntax

```
show vrrp [brief | detail | interface <INTERFACE-NAME> | interface <LAG-NAME> |
interface <VLAN-NAME> | ipv4 | ipv6 | statistics | statistics interface <INTERFACE-
NAME> | statistics interface <LAG-NAME> | statistics interface <VLAN-NAME>]
[vsx-peer]
```

Description

Shows all VRRP virtual routers information.

Command context

Manager (#)

Parameters

brief

Displays brief output of all VRRP virtual routers

Keywords used in displayed information:

- Grp: VRRP virtual router group ID.
- A-F: Address Family.
- Pri: Priority.
- Time: Uptime of VRRP virtual router since it moved out of INIT state.
- Pre: Preempt mode (Y is enabled, N if not enabled).

detail

Displays detailed output of all VRRP virtual routers

interface <INTERFACE-NAME>

Displays VRRP information for a specific interface

interface <LAG-NAME>

Displays VRRP information for a specific LAG interface

interface <VLAN-NAME>

Displays VRRP information for a specific VLAN interface

ipv4

Displays IPv4 address family

ipv6

Displays IPv6 address family

statistics

Displays VRRP statistics information for all interfaces

statistics interface <INTERFACE-NAME>

Displays VRRP statistics information for a specific interface

statistics interface <LAG-NAME>

Displays VRRP statistics information for a specific LAG interface

statistics interface <VLAN-NAME>

Displays VRRP statistics information for a specific VLAN interface

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

```
switch# show vrrp
```

VRRP is enabled

Interface 1/1/1 - Group 1 - Address-Family IPv4

State is MASTER
State duration 56 mins 57.826 secs
Virtual IP address is 10.0.0.1
Virtual MAC address is 00:00:5e:00:01:01
Advertisement interval is 1000 msec
Preemption enabled
Priority is 100
BFD is enabled
Master Router is 10.0.0.2 (local), priority is 100
Master Advertisement interval is 1000 msec
Master Down interval is unknown
Tracked object ID is 1, and state Down

Interface 1/1/2 - Group 1 - Address-Family IPv4

State is INIT (Interface Down)
State duration 45 mins 28.313 secs
Virtual IP address is no address
Virtual MAC address is 00:00:5e:00:01:01
Advertisement interval is 1000 msec
Preemption enabled
Priority is 100
BFD is disabled
Master Router is unknown, priority is unknown
Master Advertisement interval is unknown
Master Down interval is unknown

```
switch# show vrrp brief
```

Interface	Grp	A-F	Pri	Time	Owner	Pre	State	Master addr/Group addr
1/1/1	1	IPv4	100	0	N	Y	MASTER	10.0.0.2(local) 10.0.0.1
1/1/2	1	IPv4	100	0	N	Y	INIT	AF-UNDEFINED no address
1/1/2	1	IPv6	100	0	N	Y	INIT	AF-UNDEFINED no address

```
switch# show vrrp detail
```

VRRP is enabled

Interface 1/1/1 - VRRPv2 Statistics

Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0

Interface 1/1/1 - VRRPv3 Statistics

Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0

Interface 1/1/1 - Group 1 - Address-Family IPv4

State is MASTER
State duration 1 mins 35.486 secs
Virtual IP address is 10.0.0.1
Virtual MAC address is 00:00:5e:00:01:01

```
Advertisement interval is 1000 msec
Version 3
Preemption enabled
Priority is 100
BFD is enabled
Master Router is 10.0.0.2 (local), priority is 100
Master Advertisement interval is 1000 msec
Master Down interval is unknown
Tracked object ID is 1, and state Down
VRRPv3 Advertisements: sent 3931 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 3537
  IP address Owner conflicts: 0
  IP address configuration mismatch : 3537
  Advert Interval errors : 0
  Adverts received in Init state: 0
  Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 2 (Last change Mon Jun 16 11:19:36.316 UTC)
  Backup to master: 2 (Last change Mon Jun 16 11:19:39.926 UTC)
  Master to backup: 0
  Master to init: 1 (Last change Mon Jun 16 11:17:49.978 UTC)
  Backup to init: 0
```

```
switch# show vrrp interface 1/1/1
```

VRRP is enabled

```
Interface 1/1/1 - Group 1 - Address-Family IPv4
  State is MASTER
  State duration 11 mins 21.617 secs
  Virtual IP address is 10.0.0.1
  Virtual MAC address is 00:00:5e:00:01:01
  Advertisement interval is 1000 msec
  Version 3
  Preemption enabled
  Priority is 100
  BFD is enabled
  Master Router is 10.0.0.2 (local), priority is 100
  Master Advertisement interval is 1000 msec
  Master Down interval is unknown
```

```
switch# show vrrp interface lag10
```

VRRP is enabled

```
Interface lag10 - Group 1 - Address-Family IPv4
  State is MASTER
  State duration 11 mins 21.617 secs
  Virtual IP address is 10.0.0.1
  Virtual MAC address is 00:00:5e:00:01:01
  Advertisement interval is 1000 msec
  Version 3
  Preemption enabled
  Priority is 100
  BFD is enabled
  Master Router is 10.0.0.2 (local), priority is 100
  Master Advertisement interval is 1000 msec
  Master Down interval is unknown
```

```
switch# show vrrp interface vlan100
```

VRRP is enabled

```
Interface vlan100 - Group 1 - Address-Family IPv4
  State is MASTER
  State duration 11 mins 21.617 secs
  Virtual IP address is 10.0.0.1
  Virtual MAC address is 00:00:5e:00:01:01
  Advertisement interval is 1000 msec
  Version 3
  Preemption enabled
  Priority is 100
  BFD is enabled
  Master Router is 10.0.0.2 (local), priority is 100
  Master Advertisement interval is 1000 msec
  Master Down interval is unknown
```

```
switch# show vrrp statistics
```

```
VRRP is enabled
```

```
Interface 1/1/1 - VRRPv2 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
Interface 1/1/1 - VRRPv3 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
VRRP Statistics for interface 1/1/1 - Group 1 - Address-Family IPv4
  State is MASTER
  State duration 6 mins 55.006 secs
  VRRPv3 Advertisements: sent 4288 (errors 0) - rcvd 0
  VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
  Group Discarded Packets: 3856
    IP address Owner conflicts: 0
    IP address configuration mismatch : 0
    Advert Interval errors : 0
    Adverts received in Init state: 0
    Invalid group other reason: 0
  Group State transition:
    Init to master: 0
    Init to backup: 2 (Last change Mon Jun 16 11:19:36.316 UTC)
    Backup to master: 2 (Last change Mon Jun 16 11:19:39.926 UTC)
    Master to backup: 0
    Master to init: 1 (Last change Mon Jun 16 11:17:49.978 UTC)
    Backup to init: 0
```

```
Interface 1/1/2 - VRRPv2 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
Interface 1/1/2 - VRRPv3 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
VRRP Statistics for Interface 1/1/2 - Group 1 - Address-Family IPv4
  State is INIT (No Primary Group Address)
  State duration 54 mins 43.027 secs
  VRRPv3 Advertisements: sent 0 (errors 0) - rcvd 0
  VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
  Group Discarded Packets: 0
```

```
IP address Owner conflicts: 0
Invalid address count: 0
IP address configuration mismatch : 0
Advert Interval errors : 0
Adverts received in Init state: 0
Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 0
  Backup to master: 0
  Master to backup: 0
  Master to init: 0
  Backup to init: 0
```

```
VRRP Statistics for Interface 1/1/2 - Group 1 - Address-Family IPv6
State is INIT (Interface Down)
State duration 29 mins 34.508 secs
VRRPv3 Advertisements: sent 0 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 0
  IP address Owner conflicts: 0
  IP address configuration mismatch : 0
  Advert Interval errors: 0
  Adverts received in Init state: 0
  Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 0
  Backup to master: 0
  Master to backup: 0
  Master to init: 0
  Backup to init: 0
```

```
switch# show vrrp statistics interface 1/1/1
```

```
VRRP is enabled
```

```
Interface 1/1/1 - VRRPv2 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
Interface 1/1/1 - VRRPv3 Statistics
  Invalid group ID packets received : 0
  Invalid version packets received : 0
  Invalid checksum packets received : 0
```

```
VRRP Statistics for interface 1/1/1 - Group 1 - Address-Family IPv4
State is MASTER
State duration 6 mins 55.006 secs
VRRPv3 Advertisements: sent 4288 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 3856
  IP address Owner conflicts: 0
  IP address configuration mismatch : 0
  Advert Interval errors : 0
  Adverts received in Init state: 0
  Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 2 (Last change Mon Jun 16 11:19:36.316 UTC)
  Backup to master: 2 (Last change Mon Jun 16 11:19:39.926 UTC)
  Master to backup: 0
```

```
Master to init: 1 (Last change Mon Jun 16 11:17:49.978 UTC)
Backup to init: 0
```

```
switch# show vrrp statistics interface lag10
```

VRRP is enabled

Interface lag10 - VRRPv2 Statistics

```
Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0
```

Interface lag10 - VRRPv3 Statistics

```
Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0
```

VRRP Statistics for interface lag10 - Group 1 - Address-Family IPv4

```
State is MASTER
State duration 6 mins 55.006 secs
VRRPv3 Advertisements: sent 4288 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 3856
  IP address Owner conflicts: 0
  IP address configuration mismatch : 0
  Advert Interval errors : 0
  Adverts received in Init state: 0
  Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 2 (Last change Mon Jun 16 11:19:36.316 UTC)
  Backup to master: 2 (Last change Mon Jun 16 11:19:39.926 UTC)
  Master to backup: 0
  Master to init: 1 (Last change Mon Jun 16 11:17:49.978 UTC)
  Backup to init: 0
```

```
switch# show vrrp statistics interface vlan100
```

VRRP is enabled

Interface vlan100 - VRRPv2 Statistics

```
Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0
```

Interface vlan100 - VRRPv3 Statistics

```
Invalid group ID packets received : 0
Invalid version packets received : 0
Invalid checksum packets received : 0
```

VRRP Statistics for interface vlan100 - Group 1 - Address-Family IPv4

```
State is MASTER
State duration 6 mins 55.006 secs
VRRPv3 Advertisements: sent 4288 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 3856
  IP address Owner conflicts: 0
  IP address configuration mismatch : 0
  Advert Interval errors : 0
  Adverts received in Init state: 0
  Invalid group other reason: 0
Group State transition:
  Init to master: 0
  Init to backup: 2 (Last change Mon Jun 16 11:19:36.316 UTC)
```

```
Backup to master: 2 (Last change Mon Jun 16 11:19:39.926 UTC)
Master to backup: 0
Master to init: 1 (Last change Mon Jun 16 11:17:49.978 UTC)
Backup to init: 0
```

shutdown

Syntax

```
shutdown
```

```
no shutdown
```

Description

Disables VRRP group operation.

The `no` form of this command enables VRRP group operation.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling VRRP group operation:

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# shutdown

switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no shutdown
```

timers advertise

Syntax

```
timers advertise <ADVERTISE-IN-MILLISECONDS>
```

```
no timers advertise
```

Description

Sets the advertisement interval in ms (100-40950). The default value is 1000. Advertisement interval can be configured in multiples of 1,000 ms.

The `no` form of this command sets the advertisement interval in ms to the default value of 1000.



NOTE: This release does not support sub-second timer for VRRPv3.

Command context

```
config
```


Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the advertisement interval in ms to 2000:

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# timers advertise 2000
```

Setting the advertisement interval in ms to the default value of 1000:

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# no timers advertise
```

track (VRRP group)

Syntax

```
track OBJECT-ID
```

```
no track OBJECT-ID
```

Description

Sets the track object ID (1-128) for the group. The track object is first configured globally for the interface and then attached to the VRRP virtual router.



NOTE: The track object must not track the same interface for which a VRRP group is configured.

The `no` form of this command removes the track object ID from the group.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the track object ID for the group:

```
switch(config)#interface 1/1/1
switch(config-if)#vrrp 1 address-family ipv4
switch(config-if-vrrp)# track 1
```

Removing the track object ID from the group:

```
switch(config)#interface 1/1/1
switch(config-if)#vrrp 1 address-family ipv4
switch(config-if-vrrp)# no track 1
```

track (VRRP virtual router)

Syntax

```
track <OBJECT-ID>
```

```
no track <OBJECT-ID>
```

Description

Configures a track object that can be associated with an interface. A change in interface state will then affect the priority of a VRRP group. By default, no interface is associated to a track object, so state is down.

The `no` form of this command deletes a tracked object for an interface. If it is not associated with a VRRP virtual router, a track object cannot be deleted.



NOTE: Track cannot be configured by using port with no routing.

Command context

config

Parameters

<OBJECT-ID>

Specify the track object ID value. Range: 1 to 128.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring a tracked object:

```
switch(config)# track 1
```

Deleting a tracked object:

```
switch(config)# no track 1
```

track by

Syntax

```
track by OBJECT-ID
```

```
no track by OBJECT-ID
```

Description

Specifies an interface to be tracked when changes in the state of the interface affect the priority of a VRRP group. Once track is associated with an interface, the track state reflects the interface forwarding state.

The `no` form of this command removes an interface from tracking, affecting VRRP states of any interfaces associated with VRRP groups.



NOTE: The vlan interface 1 is always tracked.

Command context

config

Parameters

<OBJECT-ID>

Specifies the track object ID value. Range: 1 to 128.

Authority

Administrators or local user group members with execution rights for this command.

Example

Specifying an interface to be tracked:

```
switch (config)# interface 1/1/1
switch (config-if)# track by 1
```

version

Syntax

version <VERSION-NUMBER>

Description

Sets the protocol version for the VRRP group. Version change is allowed only for the IPv4 address-family. The default value is 2, which supports IPv4 with minimum 1 second advertisement interval. Value 3 supports IPv4 and IPv6 with minimum 1 second advertisement interval.

Command context

config

Parameters

<<VERSION-NUMBER>>

Specifies the VRRP protocol version. Possible values: 2 or 3. The default value is 2, which supports IPv4 with a minimum 1 second advertisement interval.

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the protocol version for the VRRP group to 3:

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# version 3
```

vrrp

Syntax

vrrp <VRID> address-family {ipv4 | ipv6}

no vrrp <VRID> address-family {ipv4 | ipv6}

Description

Creates a VRRP group and establishes VRRP group configuration context.

- A maximum of 8 IPv4 and 8 IPv6 VRRP groups are supported on an interface.
- A maximum of 256 VRRP groups is supported on a router. The groups can be IPv4 or IPv6 as per first come first serve basis.
- A maximum of 8 different IPv4 and 8 different IPv6 VRRP groups can be used among 1-255 VRRP groups. Achieve a scale of 256 VRRP groups by using the same VRRP group ID across different interfaces.

The `no` form of this command deletes a VRRP group.

Command context

`config`

Parameters

<VRID>

Selects the VRRP router ID value. Range: 1 to 255.

address-family [IPv4 | IPv6]

Specifies which address family to use, IPv4 or IPv6.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)#

switch(config-if-vrrp)# no vrrp 1 address-family ipv4
```

Virtual Routing and Forwarding overview

Virtual Routing and Forwarding (VRF) allows a physical router to be partitioned into multiple Virtual Router Forwarding instances (VRDs). The control and data plane are isolated in each VRF. This isolation keeps the traffic from crossing VRFs therefore multiple different routing tables can coexist within the same physical L3 switch/router.

Route leaking allows for routes to “leak” or be distributed, across multiple VRFs in a controlled manner. Two methods to leak routes from one VRF to another are:

- Static Routes
- MP-BGP (or Dynamic Route leaking using route-targets).

Static VRF route leak

Static VRF route leaking between two VRFs is allowed using an IP route configuration command. To accomplish the route leak:

- The routes being leaked to other VRFs must be present in the source VRF.
- When a route leak has been accomplished and is therefore present in the source VRF, that route can then be leaked to other VRFs.

Dynamic VRF route leak

Dynamic VRF route leaking is allowed between two or more user configured VRFs. Dynamic VRF is configured using export and import route-targets. Multiprotocol BGP (MP-BGP) is used to leak routes between the Dynamic VRFs. To configure MP-BGP, a separate address-family must be configured within BGP for each VRF, and simply redistributes routes (OSPF, static or connected) within that VRF for the routes to be leaked.

Dynamic VRF route leak restrictions and limitations

- Route leaking is supported only between user configured VRFs.
- Routes cannot be leaked from a nondefault VRF to a default VRF (or global VRF) and conversely.
- MP-BGP is used only to leak IPv4/IPv6 routes between nondefault VRFs.
- MP-BGP can be used to advertise the IPv4 or IPv6 routes to the BGP neighbors.
- BGP neighbors can be configured using IPv4 or IPv6 addresses.
- Multicast route leaking is not supported.
- Route filtering on leaked routes cannot be performed, when BGP is used as routing protocol.

- Extended Community cannot be sent to the neighbors with the prefix.
- Max supported Dynamic leaked routes per system is 16K.

Procedure to leak routes between nondefault VRFs

Follow the procedure to leak routes between nondefault VRFs:

Procedure

1. Create the VRF instances and enter the VRF context.
2. Configure the route distinguisher (RD) for each VRF.
3. Configure the appropriate address-family for each VRF.
4. Configure the BGP route targets.
5. Associate the VRFs to the interfaces.
6. Configure IP addresses on the interface.
7. Enable required IGP protocol (Ex: OSPF) on the VRF.
8. Configure multiprotocol BGP.
9. Redistribute the routes (OSPF, static or connected routes) to be leaked into BGP.

Multiprotocol BGP (MP-BGP) will be used to leak routes between VRFs. To configure MP-BGP, users must configure a separate address-family within BGP for each VRF, and simply redistributes routes (OSPF, static or connected) within that VRF for the routes to be leaked.

IVRF commands

address-family

Syntax

```
address-family [<AFI> | <SAFI>] [ipv4 | ipv6] [unicast]
```

```
[no] address-family <AFI> | <SAFI>
```

Description

Initializes the appropriate address-family and enters address-family configuration mode for IPv4 or IPv6. The unicast option is available to configure the subaddress family identifier.

The `no` form of the command removes the association of the specified address-family. The address-family specific routes that are leaked from this VRF will be withdrawn.

Command context

```
config-vrf
```

Parameters

AFI

Required: Specifies address family identifier.

SAFI

Required: Specifies subaddress family identifier. |

ipv4

Optional: IPv4 address family

ipv6

Optional: IPv6 address family

unicast

The subaddress family identifier. When the unicast option is used, the command context changes to `config-vrf-af-ipv4-uc#`.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Address family command for IPv4 unicast:

```
switch(config-vrf)#address-family ipv4 unicast
switch(config-vrf)#no address-family ipv4 unicast
```

Address family command for IPv6 unicast:

```
switch(config-vrf)#address-family ipv6 unicast
switch(config-vrf)#no address-family ipv6 unicast
```

ip|ipv6 vrf

Syntax

```
[ip|ipv6] route <PREFIX> <SRC-VRF-LOCAL-IFACE>
<SRC-VRF-NEXTHOP-IP> vrf <DST-VRF-NAME>

no [ip|ipv6] route <PREFIX> <SRC-VRF-LOCAL-IFACE>
<SRC-VRF-NEXTHOP-IP> vrf <DST-VRF-NAME>
```

Description

The IP/IPv6 route command sets the subnet mask, the reachable network interface, the next-hop IP for the reachable network, and the VRF route leak destination.

Command context

config

Parameters

<PREFIX>

The subnet mask (prefix of the network).

<SRC-VRF-LOCAL-IFACE>

The interface which is reachable by the network.

<SRC-VRF-NEXTHOP-IP>

The next-hop IP for the reachable network.

<DST-VRF-NAME>

The VRF route leak destination.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Using the command, leak the named route “Blue” VRF, using prefix 100.0.0.0/24 which is reachable by the next-hop IP 20.0.0.1 on the interface 1/1/1 from VRF “Red”:

```
switch(config)# ip route 100.0.0.0/24 1/1/1 20.0.0.1 vrf blue
```

IPv6 route source interface

Syntax

```
ipv6 route <IPv6 ADDR/MEMBER><IPv6 ADDR> source <INTERFACE>  
[<DISTANCE> | <VRF INSTANCE-NAME>]
```

```
no ipv6 route
```

Description

Creates a route leak between the source VRF and destination VRF. Using the static method, the route must first be added to the destination VRF. The route is added to the local interface of the source VRF with a next-hop interface. The existing IPv6 route command takes the source interface only when next-hop IP is link-local. To support VRF route leaking for global IPv6 unicast addresses, the command takes next-hop interface information along with next-hop IP regardless of next-hop IP is link-local or not.

- Users must provide both the next-hop IP and the interface information to leak the global unicast IPv6 network routes (route that is not directly reachable).
- The next-hop IP information is not required to leak connected global unicast IPv6 routes (route that is directly reachable).

The [no] form of command deletes the static VRF leaked route.

Command context

```
config
```

Parameters

<IPv6 ADDR/MEMBER>

Required: IPv6 IP-Address route destination.

<IPv6 ADDR>

Required: IPv6 route destination

<INTERFACE>

Required: The outgoing interface. Use the format member/slot/port (for example, 1/3/1).

<DISTANCE>

Optional: administrative distance of static route

<VRF INSTANCE-NAME>

Optional: VRF instance

Options**blackhole**

Discard packets to the destined route silently.

reject

Discard packets to the destined route and return ICMP error to the sender.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configures a route leak between the source VRF and destination VRF:

```
switch(config)# show runn
Current configuration:
!
vrf blue
    vrf green
vrf red
!

vlan 1
    interface 1/1/1
    no shutdown
    vrf attach red
ip address 2000::1/64
    interface 1/1/2
    no shutdown
    vrf attach green
ip address 3000::1/64
    interface 1/1/3
    no shutdown
    vrf attach blue
ip address 4000::1/64

switch(config)# ipv6 route 5000::0/64 3000::2 source 1/1/2 vrf red
switch(config)# ipv6 route 6000::0/64 3000::3 source 1/1/2 vrf blue

switch(config)# show runn
Current configuration:
!
vrf blue
    vrf green
vrf red
!

vlan 1
```

```

interface 1/1/1
no shutdown
vrf attach red
ip address 2000::1/64
interface 1/1/2
no shutdown
vrf attach green
ip address 3000::1/64
interface 1/1/3
no shutdown
vrf attach blue
ip address 4000::1/64

ipv6 route 5000::0/64 3000::2 source 1/1/2 vrf red
ipv6 route 6000::0/64 3000::3 source 1/1/2 vrf blue

switch(config)# no ipv6 route 5000::0/64 3000::2 source 1/1/2 vrf red
switch(config)# no ipv6 route 6000::0/64 3000::3 source 1/1/2 vrf blue

switch(config)# show runn
Current configuration:
!
vrf blue
vrf green
vrf red
!

vlan 1
    interface 1/1/1
    no shutdown
vrf attach red
    ip address 2000::1/64
    interface 1/1/2
    no shutdown
vrf attach green
ip address 3000::1/64
    interface 1/1/3
    no shutdown
vrf attach blue
    ip address 4000::1/64

```

ip route interface

Syntax

```

ip route <IPv4 ADDR/MEMBER> interface <IPv4 ADDR>
[<DISTANCE> | <VRF INSTANCE-NAME>]

```

```
no ip route
```

Description

Creates a route leak between the SRC-VRF and DST-VRF. Using the static method, the route must first be added to the destination VRF. The route is added to the local interface of the source VRF with a next-hop interface. The existing IP route command can then take either the next-hop IP or the next-hop interface.

- `INTERFACE` refers to the outgoing interface in an m/s/p format
- Users must provide both the next-hop IP and the interface information to leak network routes.
- The next-hop IP information is not required to leak connected routes.

The `[no]` form of command deletes the static VRF leaked route.

Command context

`config`

Parameters

<IPv4 ADDR/MEMBER>

Required: IPv4 IP-Address route destination.

<IPv4 ADDR>

Required: IPv4 route destination

<DISTANCE>

Optional: administrative distance of static route

<VRF INSTANCE-NAME>

Optional: VRF instance

Options

blackhole

Discard packets to the destined route silently.

reject

Discard packets to the destined route and return ICMP error to the sender.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configures a route leak between the SRC-VRF and DST-VRF:

```
switch(config)# show runn
Current configuration:
!
vrf blue
    vrf green
vrf red
!

vlan 1
    interface 1/1/1
    no shutdown
    vrf attach red
ip address 10.0.0.1/24
    interface 1/1/2
    no shutdown
    vrf attach green
ip address 20.0.0.1/24
```

```

    interface 1/1/3
        no shutdown
    vrf attach blue
    ip address 40.0.0.1/24

switch(config)# ip route A.B.C.D/M IPv4 route destination
switch(config)# ip route A.B.C.D/M IPv4 route destination
switch(config)# ip route 30.0.0.0/24 A.B.C.D Nexthop IPv4 address
switch(config)# ip route 30.0.0.0/24 1/1/2 A.B.C.D Nexthop IPv4 address
switch(config)# ip route 30.0.0.0/24 20.0.0.2 vrf green
switch(config)# ip route 30.0.0.0/24 1/1/2 20.0.0.2 vrf red
switch(config)# ip route 50.0.0.0/24 1/1/2 20.0.0.2 vrf blue
switch(config)# ip route 50.0.0.0/24 20.0.0.2 vrf green
switch(config)# ip route 60.0.0.0/24 1/1/2 vrf red

switch(config)# show runn
Current configuration:
!
vrf blue
    vrf green
vrf red
!

vlan 1
    interface 1/1/1
        no shutdown
        vrf attach red
    ip address 10.0.0.1/24
    interface 1/1/2
        no shutdown
        vrf attach green
    ip address 20.0.0.1/24
    interface 1/1/3
        no shutdown
        vrf attach blue
    ip address 40.0.0.1/24
    ip route 30.0.0.0/24 1/1/2 20.0.0.2 vrf red
    ip route 50.0.0.0/24 1/1/2 20.0.0.3 vrf blue
    ip route 60.0.0.0/24 1/1/2 vrf red

switch(config)# no ip route 30.0.0.0/24 1/1/2 20.0.0.2 vrf red
switch(config)# no ip route 50.0.0.0/24 1/1/2 20.0.0.3 vrf blue
switch(config)# no ip route 60.0.0.0/24 1/1/2 vrf red

switch(config)# show runn
Current configuration:
!
    vrf blue
    vrf green
vrf red
!

vlan 1
    interface 1/1/1
        no shutdown
        vrf attach red

```

```
        ip address 10.0.0.1/24
interface 1/1/2
no shutdown
vrf attach green
ip address 20.0.0.1/24
interface 1/1/3
no shutdown
vrf attach blue
ip address 40.0.0.1/24
```

rd

Syntax

```
rd <AS-NUMBER:NN>
```

```
no rd <AS-NUMBER:NN>
```

Description

Configures VRF table with specified route-distinguisher value. An RD ensures uniqueness of a route between multiple VRFs.

The no form of the command will delete RD from a specified VRF table. The VRF instance goes down when RD is deleted. All routes that are exported or leaked from the deleted VRF will be withdrawn.

Command Context

configure-vrf

Parameters

<AS-NUMBER:NN>

Required: Enter an AS number and an arbitrary number.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configures VRF for RD with an AS number 100:1.

```
switch(config-vrf)# rd 100:1
```

Deletes the RD from the specified VRF.

```
switch(config-vrf)# no rd
```

Deletes the RD and AS number from the specified VRF.

```
switch(config-vrf)# no rd 100:1
```

redistribute

Syntax

```
redistribute <protocol> [route-map <route-map-name>]
```

```
[no] redistribute <protocol> [route-map <route-map-name>]
```

Description

Specifies the protocol routes to redistribute to BGP VRF context. Any routes existing in the BGP VRF context are leaked as a VPNv4 or VPNv6 prefixes to other VRFs based on BGP route-targets.

The **no** form of this command removes the protocol.

Command context

config

Parameters

redistribute

Required: redistributes routes from another routing protocol.

connected

Optional: redistribute directly attached networks.

ospfv3

Optional: redistributes OSPFv2 routes.

static

Optional: redistributes static routes.

route-map

Optional: applies route map policy for redistribution.

Authority

Administrators or local user group members with execution rights for this command.

Example: OSPFv2 redistribute routes to BGP instance

The following is an example of redistributing OSPFv2 routes to a BGP `vrf cust_a` instance by creating a router BGP instance for `cust_a`.

1. Creating the router BGP instance for `cust_a`.

```
switch(config)# router bgp 1  
switch(config-router) # vrf cust_a
```

2. Redistributing the router to BGP.

```
switch(config-router-bgp) # redistribute ospf
```

Example: OSPFv3 redistribute routes to BGP instance

The following is an example of redistributing OSPFv3 routes to a BGP `vrf cust_a` instance by creating a router BGP instance for `cust_a`.

1. Creating the router BGP instance for `cust_a`

```
switch(config)# router bgp 100  
switch(config-router) # vrf cust_a
```

2. Configuring the address family IPv6 unicast to the router

```
switch(config-router-bgp) # address-family ipv6 unicast
```

3. Redistributing the router to OSPFv3

```
switch(config-router-ipv6-uc) # redistribute ospfv3
```

4. Redistributing the router configured with ipv6-af-us to OSPFv3

```
switch(config-router-bgp-vrf-ipv6-af-uc) # redistribute ospfv3
```

route-target

Syntax

```
route-target [import | export | both] <AS-NUMBER:NN>
```

```
no route-target [import | export | both] <AS-NUMBER:NN>
```

Description

BGP route targets are extended BGP communities that identify the VPNv4 or VPNv6 routes that are associated with a VRF. This command specifies the route targets used on the import or export of the routes to other VRFs. Multiple route targets can be associated to a VRF.

The `no` form of the command removes the association.

Command context

Manager (#)

Parameters

import

Specifies the RTs imported to the VRF. Import or export or both required Literal Specifies the route-target type.

export

Specifies the RT on VPNv4 or VPNv6 prefixes that are leaked to other VRFs.

both

Specifies the RT for both export and import types.

<AS-NUMBER:NN>

Specifies an AS number and an arbitrary number for the RT value.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configures the route-target for export. Removes the configuration for export.

```
switch(config-vrf-ipv4-af-uc) # route-target export 100:1
```

```
switch(config-vrf-ipv4-af-uc) # no route-target export 100:1
```

Configures the route-target for import. Removes the configuration for import.

```
switch(config-vrf-ipv4-af-uc)# route-target import 100:2
```

```
switch(config-vrf-ipv4-af-uc)# no route-target import 100:2
```

Configures the route-target for both import and export. Removes the configuration for import and export.

```
switch(config-vrf-ipv4-af-uc)# route-target both 100:3
```

```
switch(config-vrf-ipv4-af-uc)# no route-target both 100:3
```

router bgp

Syntax

```
router bgp <AS-NUMBER>
```

```
no router bgp <AS-NUMBER>
```

Description

This command configures the BGP instance on the router, configures the AS (Autonomous System) the router belongs to, and enters into the BGP router configuration mode. Only a single BGP AS number can be assigned for the entire system.

The `no` form of the command deletes the BGP instance from the router.

Command context

```
config
```

Parameters

AS-NUMBER

Required: Specifies a 4-byte AS number in the range 1-4294967295 in integer format or from 0.1-65535.65535 in dotted format.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring the BGP instance with the AS number:

```
switch(config)# router bgp 100
```

Deleting BGP configurations:

```
switch(config)# no router bgp 100
```

```
This will delete all BGP configurations on this device.  
Continue (y/n)?
```

router bgp vrf

Syntax

```
router bgp <AS-NUMBER:NN> [vrf <VRF-NAME>]
```

```
[no] router bgp <AS-NUMBER:NN> [vrf <VRF-NAME>]
```


Description

This command configures VRF for the BGP instance.

The `no` form of this command removes the configuration.

Command context

`config`

Parameters

<AS-NUMBER:NN>

Specifies an AS number and an arbitrary number for the RT value.

<VRF-NAME>

String VRF name for the VRF.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Use the command `vrf vrf-name` within the router BDG context.
- `address-family {ipv4 | ipv6}` nodes are only supported within the VRF context.
- `address-family {ipv4 | ipv6}` nodes are required to redistribute the OSPF static/connected IPv4 or IPv6 routes.

Examples

Configure the VRF for customer A, on the BGP instance 100:

```
switch(config)# router bgp 100
switch(config)# vrf cust_a
```

show bgp vpn unicast

Syntax

```
show bgp [vrf <VRF-NAME>][{ipv4 unicast | ipv6 unicast| ipv4 unicast| all unicast}] [vsx-peer]
```

Description

Shows the BGP-VPN per VRF routes with additional route information like RD and extended community route targets.

Command context

Operator (>)

Parameters

Displays the BGP neighbor information for the specified VRF.



NOTE: By default the default_vrf BGP instance information is displayed if the VRF is not specified.

unicast

Selects the subaddress family identifier

all

Displays VPNv4 address family routes for all VRFs

vrf

Displays VPNv4 address-family routes for specified VRF

vpn-addr-family

Required: Literal Select the VPNv4 or VPNv6 address family

vrf-name

Required: Literal or string. Specify **all** or *vrf-name*.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Required use of *vpn-addr-family*:

```
switch# show bgp <vpn-addr-family> unicast {all | [vrf <vrf-name> | A.B.C.D/M]}
```

Show BGP VRF ipv4 unicast routes for vrf-name

```
switch# show bgp vrf Red ipv4 unicast
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: Red
Local Router-ID 172.16.3.1

Network          Nexthop          Metric  LocPrf  Weight Path
Route Distinguisher: 65000:1
*> 172.16.0.0/24  0.0.0.0           0       100     32768    ?
*> 172.16.1.0/24  0.0.0.0           0       100     32768    ?
*> 172.16.2.0/24  172.16.0.2        0       100     32768    ?
*> 172.16.3.0/24  172.16.0.3        0       100     32768    ?

Total number of entries 4
```

```
switch# show bgp vrf Green ipv4 unicast
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: Green
Local Router-ID 172.17.2.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65000:2					
*> 172.17.0.0/24	0.0.0.0	0	100	32768	?
*> 172.17.1.0/30	0.0.0.0	0	100	32768	?
*> 172.17.2.0/24	172.17.0.2	0	100	32768	?

Total number of entries 3

```
switch# show bgp vrf Blue ipv4 unicast
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: Blue
Local Router-ID 172.18.3.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65000:3					
*> 172.18.0.0/24	0.0.0.0	0	100	32768	?
*> 172.18.1.0/30	0.0.0.0	0	100	32768	?
*> 172.18.3.0/24	172.18.0.3	0	100	32768	?

Total number of entries 3

```
switch# show bgp vrf Shared ipv4 unicast
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: Shared
Local Router-ID 192.168.99.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65000:99					
*> 192.168.99.0/24	0.0.0.0	0	100	32768	?

Total number of entries 1

Show BGP VRF ipv6 unicast routes for vrf-name:

```
switch# show bgp vrf Red ipv6 unicast
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete

VRF: Red
Local Router-ID 172.16.3.1
```

Network	Nexthop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65000:1					
* 2001:100:1:1000::/56					
2001:100:1:1000::72a	0	0	200	?	
*> 2001:100:1:1000::/56					
::	0	100	32768	?	
* 2001:100:1:2000::/56					
::FFFF:200.10.10.1 0	0	100	32768	?	

Total number of entries 3

Show BGP VRF routes for all vrf's and all address-families:

```
switch# show bgp all-vrf all
      Status codes: s suppressed, d damped, h history, * valid, > best, =
multipath
      i internal, e external, S Stale, R Removed
      Origin codes: i - IGP, e - EGP, ? - incomplete
```

VRF: Blue

Local Router-ID 172.18.3.1

Address-family: IPv4 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
*> 172.18.0.0/24	0.0.0.0	0	100	32768	?
*> 172.18.1.0/30	0.0.0.0	0	100	32768	?
*> 172.18.3.0/24	172.18.0.3	0	100	32768	?

Total number of entries 3

Address-family: IPv6 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
---------	---------	--------	--------	--------	------

Total number of entries 0

VRF: Green

Local Router-ID 172.17.2.1

Address-family: IPv4 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
*> 172.17.0.0/24	0.0.0.0	0	100	32768	?
*> 172.17.1.0/30	0.0.0.0	0	100	32768	?
*> 172.17.2.0/24	172.17.0.2	0	100	32768	?

Total number of entries 3

Address-family: IPv6 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
---------	---------	--------	--------	--------	------

Total number of entries 0

VRF: Red

Local Router-ID 172.16.3.1

Address-family: IPv4 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
*> 172.16.0.0/24	0.0.0.0	0	100	32768	?
*> 172.16.1.0/24	0.0.0.0	0	100	32768	?
*> 172.16.2.0/24	172.16.0.2	0	100	32768	?
*> 172.16.3.0/24	172.16.0.3	0	100	32768	?

Total number of entries 4

Address-family: IPv6 Unicast

Network	Nexthop	Metric	LocPrf	Weight	Path
* 2001:100:1:1000::/56					

```

      2001:100:1:1000::72a      0      0      200      ?
*> 2001:100:1:1000::/56
      ::                        0      100     32768     ?
* 2001:100:1:2000::/56
      ::FFFF:200.10.10.1 0      0      100     32768     ?

Total number of entries 3

VRF: Shared
Local Router-ID 192.168.99.1

Address-family: IPv4 Unicast
-----
Network          Nexthop      Metric   LocPrf   Weight   Path
*> 192.168.99.0/24  0.0.0.0      0        100     32768     ?

Total number of entries 1

Address-family: IPv6 Unicast
-----
Network          Nexthop      Metric   LocPrf   Weight   Path

Total number of entries 0

```

show bgp info vrf

Syntax

```
show bgp info vrf <vrf-name> [vsx-peer]
```

Description

Displays BGP route-targets information for specified VRF.

Command context

Operator (>) or Manager (#)

Parameters

info

Display BGP RT information.

vrf-name

Required string VRF name for the vrf.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Show bdg vrf information.

```
switch# show bgp info vrf red
VRF : red
VRF RD : 100:1

Address-family IPv4 unicast info
Redistribution : ospf
Export RT list : 100:1 100:2
Import RT list : 100:3

Address-family IPv6 unicast info
Redistribution : connected
Export RT list : 100:11 100:12
Import RT list : 100:15
```

show ip route vrf

Syntax

```
show ip route vrf <vrf-name> [vsx-peer]
```

Description

Shows route information for specified VRF.

Command context

Operator (>) or Manager (#)

Parameters

vrf-name

Required: string VRF name for the VRF.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show ip route vrf green

Displaying ipv4 routes selected for forwarding
'[x/y]' denotes [distance/metric]
 10.0.0.0/24, vrf green
  via 20.0.0.1[vrf red], [1/0], static
 30.0.0.0/24, vrf green
  via 1/1/2, [0/0], connected
30.0.0.2/32, vrf green
  via 1/1/2, [0/0], local
```

```
60.0.0.0/24, vrf green
via 1/1/1[vrf red], [1/0], static
```

show ipv6 route vrf

Syntax

```
show ipv6 route vrf <VRF-NAME> [vsx-peer]
```

Description

Shows the route information for specified VRF.

Command context

Operator (>) or Manager (#)

Parameters

<VRF-NAME>

Required: String VRF name for the VRF.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying ipv6 routes selected for forwarding:

```
switch# show ipv6 route vrf red
'[x/y]' denotes [distance/metric]
 1000::/64, vrf red
 via 1/1/1[vrf green], [0/0], connected
 1000::1/128, vrf red
 via 1/1/1[vrf green], [0/0], local
 3005::/64, vrf red
 via 1000::2[vrf green], [2/0], static
```

vrf

Syntax

```
vrf <VRF-NAME>
```

```
no vrf <VRF-NAME>
```

Description

This command configures a named VRF instance <VRF-NAME>. After creating the VRF instance, the command prompt enters into the VRF configure context.

The `no` form of the command deletes the VRF instance from the router.

Command context

config

Parameters

<VRF-NAME>

Required: Specifies the VRF name. Length: Up to 32 alphanumeric characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring VRF for a named customer:

```
switch(config) # vrf cust_A
```

Removing the VRF configuration with the name **test**:

```
switch(config) # no vrf test
```


Overview

Policy Based routing (PBR) lets you manipulate the path of a packet based on the various attributes of the packet. Packets that can be manipulated by PBR are packets that are already routing through the system at Layer 3, with a destination IP address that is on a network other than the packet ingress Layer 3 interface. PBR is an extension of the existing classifier policy system where traffic to be manipulated is matched by a classifier **class**, and policy **actions** to be executed on the matching traffic. Matching traffic with the same destination can be routed over different paths so that different types of traffic, such as VoIP or traffic with special security requirements, can be better managed.



NOTE: PBR ability to influence a packet path is limited to the current router in that the next router to which the traffic is redirected makes an independent decision on where to forward traffic to next, if anywhere. Packets not matched by a PBR policy entry are not affected and take routes specified in the system route table.

PBR actions

PBR can alter a packet path through the following PBR actions:

- **interface null:** equivalent to the policy **drop** policing action. Any packets matching the class criteria for that policy entry will be dropped and not routed any further.
- **interface tunnel:** allows for specifying a GRE, 6in4 or 6in6 tunnel as the outbound interface for all matching packets. The tunnel must exist before configuring. Packets sent into the tunnel interface egress at the router at the endpoint of the tunnel. If the tunnel is misconfigured or down the traffic may be lost.
- **nexthop:** allows for overriding the routing table's longest prefix match next-hop router for matching packets. If no such routing table entry exists for matching packets, (default or not) this action still affects matching packets.
- **default-nexthop:** allows for specifying a next-hop router for matching packets when there is no longest prefix match for those packets in the routing table. Such a default-nexthop overrides a system default route if already configured and also applies if there is no system default route.



NOTE:

- Next-hop and default-nexthop facilitate routing matching packets where they otherwise might not be, due to the absence of routing table entries.
- Unlike next-hop, default-nexthop only applies if there is no destination lookup match in the main routing table for matching packets.

PBR policy action and action list

Entries in a policy use a class to specify the criteria on what packets to match and a set of one or more policy actions to take on the matching packets such as dscp or mirror.

PBR is configured for use as another such policy action in a policy entry by using the `pbr` keyword followed by the name of a user configured PBR action list. This list may contain up to eight PBR action entries which

are listed according to priority by sequence number. The **PBR actions** section lists the different action entry choices.

In a given action list applied to an interface, zero entries may be available (that is, reachable). Of the entries that are available, only one can be active for that action list in a given VRF. Due to real-time changes in the network operating environment, list entry availability can change at any moment. At any given time, the highest priority available entry in the list is selected as the active PBR action for that list. If the active entry becomes unavailable, the next highest available entry is automatically selected and replaces that now-unavailable entry as the active entry. If a higher priority entry (other than the current active entry) becomes available, this entry is selected and replaces the current active entry, as the new active entry (even though that current entry is still available). If no entries are available or the only available entry becomes unavailable, routing occurs based on the routing table and is not influenced by PBR.



NOTE: The **interface null** action is always considered available. If there are no higher priority PBR actions available, it will be selected as the active entry when configured in a PBR action list, and applied to an interface.

PBR action list maximum entries

There is a limit of eight entries per action list and a limit of 512 unique entries across the entire system. This value restricts the sum of all unique next-hops, default-nexthops, tunnel interfaces, and null interfaces applied to all Layer 3 interfaces across all VRFs. A given set of entries that are unique in one VRF - count against this limit if the set is applied in another VRF.

For example, a given next-hop specified in an action list applied to an interface counts against the limit once. If that same next-hop address is entered in a different action list and applied to another interface in the same VRF, this next-hop is not unique in the VRF and does not affect the limit further. If one of those action lists (that specify the address) is applied to an interface in a different VRF, the entry will affect the limit since addresses are unique to each VRF.

Another example is if two PBR action lists with eight unique next-hop entries (for a total of 16 unique next-hops) are configured as PBR actions in a policy. If that policy is applied to 32 route-only ports (each of which is in a different VRF), the entry supply will be exhausted.



NOTE: The specified capabilities and capacities limit for unique VRFs is 33.

IP versions in an action list

Configuring PBR actions of different IP versions in the same PBR action list is not supported. This limitation applies to the next-hop and default-nexthop actions specifically. If an action of one IP version is already configured, configuring an action with a different IP version in the same action list will be blocked. This limitation applies regardless of whether the action list is part of a policy, applied to an interface or whether any of the entries are selected or not.

Furthermore, use cases similar to the following are not supported:

- Applying a PBR action list with entries of one IP version (for example IPv4 next-hop) to an interface of the other IP version (for example a route-only port with an IPv6 address only).
- Creating a policy entry with a class of one IP version and a PBR action list with entries of the other IP version.

Configuration mismatches while not prevented are not supported and behavior is not defined. At best, traffic will not reach its intended destination.

Specifying valid next-hop and default-nexthop addresses

PBR does not support remote routers as next-hop or default-nexthop routers (a.k.a recursive). PBR only supports routers that are on a directly connected network. During configuration you may specify any IPv4 or IPv6 address as a next-hop or default-nexthop, however only those addresses reachable through a directly connected network will be installed as the active PBR entry.



CAUTION: If a specified address is reachable from the router, it must be on a directly connected subnet or it will not become active.

Hardware path PBR versus software path PBR

Traffic to be routed that cannot be handled by the switching ASIC (for example IPv4 packets with IP options), is handled by the switch operating system kernel routing software.

The characteristics of network usage determine the proportion of traffic handled by software path PBR relative to the hardware path. Under normal conditions, software path PBR is likely to range from a fraction of a percent of all traffic to almost none.

Packets with IP options are sent to the hardware ASIC as well as the CPU. Since the software path PBR actions are applied to the packets in the CPU, a default entry is programmed in the hardware to take no action on those packets. This hardware entry count can be seen in diagnostics with the command `diag-dump policy basic`.

Hardware versus software path for default-nexthop action

When a policy containing an entry with PBR is applied to an interface and the active action for that entry action list is 'default-nexthop', packets that match the class criteria and have no explicit route table hit (that is, no destination address longest prefix match, a.k.a 'route-miss'), they are forwarded to the specified active PBR default-nexthop. This is true for packets that follow the hardware and software paths through the switch.

There is a difference between PBR hardware and software path behavior when a route table hit occurs for class-matched packets (when the policy entry is a PBR default-nexthop).

Hardware path match criteria for a PBR policy entry with default-nexthop is extended to include the route table miss along with the class qualifiers, resulting in a **policy entry hit** for that policy entry. Conversely when there is a route table hit, the result is a policy entry miss in hardware path. When a policy entry miss occurs, policy processing moves on to the next entry in the policy and takes whatever action is specified, if any exist. This can include a different PBR routing action including interface null or no PBR action at all, for example.

In software path, the class match criteria is the only criteria required to achieve a policy entry hit. When that occurs, policy processing will stop. When there is a class match and a route table hit (with PBR action default-nexthop), the packet is forwarded according to that route table entry, not the PBR default-nexthop entry, nor is it influenced by any subsequent policy entries.

This difference in behavior is due to a limitation in the software path matching and routing engine, relative to hardware.

Routing packet	System route miss	System route hit
Hardware	Take PBR default-nexthop route.	Policy entry miss, continue policy processing with next entry, if present. No further matches result in system default route being used, if present.
Software	Take PBR default-nexthop route.	Policy entry hit is overridden by system route hit. System route entry used, policy processing stops.

Software path and system default route

AOS-CX default behavior (for packets with destination networks that lack entries in the system route table with a reachable next-hop address), sends packets to the system CPU for special handling, if any are specified (for example, sending an ICMP unreachable reply message).

A side effect of this CPU routing is that it has higher priority than an applied PBR next-hop action. Even though a policy is applied with an entry that matches the traffic and specifies a PBR next-hop action which is reachable, the traffic will still be routed to the system CPU (due to the absence of a reachable next-hop in the system route table), and not through the desired PBR hardware path. With traffic routing to the system CPU, it will be properly routed by PBR software path, but it will also be rate limited by the control plane policing feature. Loss can occur at higher traffic rates.



CAUTION: The workaround for this issue is to create a default next-hop route in the system with a reachable next-hop router/host. This will result in a route hit, or reachable next-hop detected for the traffic with no further need to route traffic to the CPU. The PBR hardware path next-hop action will then occur, as desired.

PBR and VRFs

A given physical router can be partitioned into multiple virtual routers. All frontplane ports and logical interfaces are initially in the default VRF. Any subset of frontplane ports or logical interfaces can be selectively moved into a user configured VRF, to establish a routing topology that is different from the default VRF.

Policies with PBR actions can be applied to interfaces in any VRF, and PBR action lists can be used in different policies across different VRFs. The effect of applying the same Policies/PBR action lists across different VRFs depends on the IP networks and interfaces configured in the different VRFs. For example, an action list that specifies actions of 'nexthop 1.1.1.10' and 'interface tunnel gre_10' could be used as the PBR action parameter for an entry in policy_1 and also in policy_2. If policy_1 is applied to an interface in VRF 'red', which has an interface in subnet 1.1.1.0/24 but no GRE tunnel named 'gre_10', then only the 'nexthop' action will be relevant to VRF 'red'. If policy_2 (which contains the same action list), is applied to an interface in VRF 'blue' (which lacks the 1.1.1.0/24 subnet configured but does have a tunnel named 'gre_10'), then only that interface tunnel action will apply in that VRF.



NOTE:

- It is possible to configure the same subnet in different VRFs, however named tunnel interfaces can only exist in one, so in the example of a common action list, the 'next-hop' action could be relevant to both VRFs, but the 'interface tunnel' action may only be relevant to one. If VRFs are part of the router configuration, be mindful of them when creating and applying policies with PBR action lists and their entries.
- VRF Route Leaking is not supported in the current release of PBR (10.4).

PBR, ECMP, and routing protocols

When a policy with PBR actions is applied to an interface, the highest priority action from the action list is applied to matching traffic on that interface and overrides any static, ECMP or dynamically installed routes in the system.

PBR, VSX, and VLAN ACLs

Multichassis Link Aggregation Group (MCLAG) with VSX is a high-availability feature where a switch containing LAG members is connected to multiple switches to allow for node-level redundancy on that link. If one of the other switches goes down the LAG remains up and can continue to carry all the LAG traffic, bandwidth permitting.

A LAG (and an MCLAG) can be a member of a VLAN and a Layer 3 VSI (virtual switch interface), which can be created for that VLAN for the purposes of routing a policy with Layer 3 specific actions (that is, PBR), which can therefore be applied to that interface to influence routing decisions for matching Layer 3 packets on the MCLAG; like any other route-only port or VSI. Under such a configuration, it is likely that it is desirable to use VLAN ACLs applied to the VLAN the MCLAG is a member of.



NOTE: There is a limit with this particular combination when the VLAN ACL specifies both IPv4 and IPv6 entries and the PBR policy has entries with IPv4 and IPv6 classes. This configuration will exhaust the switching ASIC resources and will fail to apply. To achieve a similar configuration, you may use port ACLs for IPv4 traffic (applying the ACL to all ports individually in the VLAN) while preserving the rest of the configuration.

PBR software path, VSX, and VRRP

Due to the dynamic nature of the VSX and VRRP protocols, applying a policy with PBR to VSX or VRRP Layer 3 kernel interfaces and then making further changes to those protocols may not result in expected behavior. Apply policies with PBR to VSX and VRRP Layer 3 interfaces after all changes have been made. If further changes are necessary the policy should be removed first and reapplied when configuration updates are complete.

PBR and next-hop router reachability

When a policy with PBR actions is applied to an interface, all next-hop and default-nexthop entries in the associated action list are continuously monitored for reachability. Probing occurs every 5 seconds for each such entry and guarantees that the reaction to the loss of reachability, or detection of the new reachability of any such entry will be approximately 5 seconds.

For example, if there are two reachable next-hops in an applied action list and the active next-hop (highest priority by lowest sequence number) entry becomes unreachable (loss of link or next-hop power event for example), the switch over to the lower priority next-hop will occur within approximately 5 seconds. If there is only one nexthop/default-nexthop entry and no entries of any other type in the action list, routing decisions will return to the system routing table.

Conversely if an applied action list with a single unreachable next-hop becomes reachable, the switch back to routing to that next-hop will occur when reachability of the next-hop on the network is achieved. On an action list with multiple next-hops (where the active next-hop is not the highest priority entry); once the next-hop with the higher priority becomes available, the now reachable next-hop will promote to active once new reachability on the network is achieved.

CLI errors

Message	Event
Failed to create action list	A PBR action list create operation was unable to create the OVSDB row.
PBR action list '%s' doesn't exist	A PBR action list update or delete operation was unable to locate the list OVSDB row. OR A PBR action list entry create, update, or delete operation was unable to locate its parent action list OVSDB row.
Unable to automatically set sequence number	The automatically generated sequence number for a new PBR action list entry is higher than the maximum permitted sequence number.
Unable to add PBR action list entry	A PBR action list entry create failed to create an entry row in OVSDB.
Invalid sequence number	A PBR action list entry delete did not specify the entry sequence number.
Action list entry does not exist	A PBR action list entry delete could not locate the OVSDB entry row.
Action list name required	A PBR action list delete was passed an invalid list name.
Invalid action list name length	A PBR action list name was longer than the permitted length.

PBR commands

apply policy

Syntax

```
apply policy <POLICY-NAME> routed-in
```

```
no apply policy <POLICY-NAME> routed-in
```

Description

Applies a classifier policy containing a PBR action to an interface. A policy with PBR actions is only applicable to L3/routing interfaces.

The `no` form of this command removes a classifier policy containing a PBR action from an interface.

Command context

```
config-if
```

Parameters

<POLICY-NAME>

Specifies name of the policy.

Authority

Administrators or local user group members with execution rights for this command.

Restrictions

- Only Layer 3 interfaces are valid for PBR policy application, and only in the routed inbound direction.
- If a policy with an 'interface tunnel' PBR action is applied on a Layer 3 interface in VRF 'A', and that interface tunnel is a member of VRF 'B', the interface tunnel is considered down/unavailable in this policy application in VRF 'A'.

Example

On the 6400 Switch Series, interface identification differs.

Apply policy to an interface:

```
switch(config)# interface 1/1/10
switch(config-if)# routing
switch(config-if)# apply policy pbr_policy routed-in
switch(config-if)# exit
```

pbr-action-list

Syntax

pbr-action-list <ACTION-LIST-NAME>

```
[<SEQUENCE-NUMBER>]
 {nexthop | default-nexthop} <NEXT-HOP-IP-ADDR>
 interface {null | <TUNNEL-NAME>}
```

```
no [<SEQUENCE-NUMBER>]
 {nexthop | default-nexthop} <IP-ADDR>
 interface {null | <TUNNEL-NAME>}
```

no pbr-action-list <ACTION-LIST-NAME>

Description

Creates a PBR action list or modifies its entries.

The **no** form of this command can be used to delete an action list or an individual action list entry.

Command context

config

The **pbr-action-list <ACTION-LIST-NAME>** command takes you into the **config-pbr-action-list-<ACTION-LIST-NAME>** context where you modify entries for a PBR action list.

Parameters

<ACTION-LIST-NAME>

Specifies the action list name. An action list name can be 1 to 64 alphanumeric characters.

<SEQUENCE-NUMBER>

Specifies list entry sequence number. Range: 1-4294967295

{nexthop | default-nexthop}

Selects a regular next-hop (**nexthop**) or a default next-hop (**default-nexthop**). These parameters specify the address of a next-hop router to forward traffic matched by a class under different conditions.

nexthop

Sets the next hop for routing the packet.

default-nexthop

Sets the next hop for routing the packet when there is no explicit route for its destination.

<NEXTHOP-IP-ADDR>

Specifies IPv4 or IPv6 address of the next-hop router.

interface {null | <TUNNEL-NAME>}

Selects the type of keyword interface: **null** or the tunnel interface name.

null

Specifies to drop matching traffic.

<TUNNEL-NAME>

Specifies an IP tunnel interface name through which to forward the matching traffic.

Authority

Administrators or local user group members with execution rights for this command.

Restrictions

The reachability of the next-hop routers/tunnel interfaces in the list is not guaranteed. Such reachability can change at any time due to the dynamic nature of the network environment.

Usage

Each action list may contain up to eight entries of four different entry types:

- `interface null`
- `interface tunnel`
- `nexthop`
- `default-nexthop`

List entries have a unique sequence number which, if not user specified, are automatically assigned beginning at 10 and continuing at intervals of 10 for each subsequent new list entry, for example 20, 30, and 40. Sequence numbers of any value can be specified manually, a different interval may be set, and new entries can be added to (or removed from) any location in the list at any time.

Specifying an existing sequence number causes the existing list entry to be replaced by the new details. The list entry with the lowest sequence number has the highest priority entry in the list. The sequence numbers may be renumbered with the **pbr-action-list resequence command**.

Only one next-hop router or interface from the list is used per packet matched. This router or interface is defined as the highest priority list entry that is reachable or available at the time of the traffic match. If the highest priority list entry next-hop router or tunnel interface is reachable - that list entry is chosen, the search is stopped - and the traffic is forwarded to the next-hop router or interface for the entry. If the highest priority list entry next-hop router or tunnel interface is not reachable, the next highest priority list entry reachability is determined and used if reachable, otherwise the process continues down the list. If none of the routers in the list are reachable, the packet may be dropped through the null interface entry if configured) or forwarded according to a system route table entry.



NOTE: An action list that contains a next-hop of one IP version cannot also contain an entry of another IP version. For example, an action list must contain only IPv4 or IPv6 next-hop addresses or tunnel interfaces.

Examples

The list name is included in the context prompt for easy current-list identification. Any list name over 10 characters will be truncated at 10 characters and terminated with the tilde character (~) to indicate a reduced list name display. This reduction affects the prompt display of the list name only:

```
switch(config)# pbr-action-list eighteenchars
switch(config-pbr-action-list-eighteench~)#
```

The following example creates an action list with two IPv4 next-hops, a default IPv4 next-hop, and a null interface. The example uses default sequence numbering for its list entries.

```
switch(config)# pbr-action-list test1
switch(config-pbr-action-list-test1)# nexthop 1.1.1.1
switch(config-pbr-action-list-test1)# nexthop 2.2.2.2
switch(config-pbr-action-list-test1)# default-nexthop 9.9.9.9
switch(config-pbr-action-list-test1)# interface null
switch(config-pbr-action-list-test1)# end
```

```
switch(config)# show pbr-action-list test1
```

Sequence	Name Type	Address/Interface

	test1	
10	nexthop	1.1.1.1
20	nexthop	2.2.2.2
30	default-nexthop	9.9.9.9
40	interface	null

The following example creates an action list with an IPv4 next-hop and a tunnel interface with manual sequence numbers for its entries.

```
switch(config)# pbr-action-list test2
switch(config-pbr-action-list-test2)# 6 ip default-nexthop 4.4.4.4
switch(config-pbr-action-list-test2)# 1 interface tunnel10
switch(config-pbr-action-list-test2)# end
```

```
switch(config)# show pbr-action-list test2
```

Sequence	Name Type	Address/Interface

	test2	
1	interface	tunnel10
6	default-nexthop	4.4.4.4

The following example creates an action list with two IPv4 tunnel interfaces, with default sequence numbering.

```
switch(config)# pbr-action-list test3
switch(config-pbr-action-list-test3)# interface tunnel10
switch(config-pbr-action-list-test3)# interface tunnel15
switch(config-pbr-action-list-test3)# end
```

```
switch(config)# show pbr-action-list test3
```

Sequence	Name Type	Address/Interface

	test3	
10	interface	tunnel10
20	interface	tunnel15

The following example creates an action list with two IPv6 next-hops and the null interface, with manual sequence numbers.

```
switch(config)# pbr-action-list test4
switch(config-pbr-action-list-test4)# 5 nexthop 2000:abcd::cccc:dddd
switch(config-pbr-action-list-test4)# 6 nexthop 1000:abcd::1234:5678
switch(config-pbr-action-list-test4)# 7 interface null
switch(config-pbr-action-list-test4)# end
```

```
switch(config)# show pbr-action-list test4
```

Sequence	Name Type	Address/Interface

	test4	
5	nexthop	2000:abcd::cccc:dddd
6	nexthop	1000:abcd::1234:5678
7	interface	null

pbr-action-list copy

Syntax

```
pbr-action-list <ACTION-LIST-NAME> copy <DESTINATION-ACTION-LIST-NAME>
```

Description

Copies an existing PBR action list.

Command context

config

Parameters

<ACTION-LIST-NAME>

Specifies the action list name to be copied.

<DESTINATION-ACTION-LIST-NAME>

Specifies the name of the copied action list. A destination action list name can be 1 to 64 alphanumeric characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

The following example copies test4 action list to test 5.

```
switch(config)# show pbr-action-list test4
```

Sequence	Name Type	Address/Interface

	test4	
5	nexthop	2000:abcd::cccc:dddd
6	nexthop	1000:abcd::1234:5678
7	interface	null

```
switch(config)# pbr-action-list test4 copy test5
```

```
switch(config-pbr-action-list-test4)# show pbr-action-list test5
```

Sequence	Name Type	Address/Interface

	test4	
1	nexthop	2000.abcd::cccc.dddd
11	nexthop	1000.abcd::1234.5678
21	interface	null

pbr-action-list resequence

Syntax

```
pbr-action-list <ACTION-LIST-NAME> resequence <STARTING-SEQUENCE-NUMBER> <INCREMENT>
```

Description

Renumbers the entries in an action list. The list entry with the lowest sequence number has the highest priority entry in the list.

Command context

```
config
```

Parameters

<ACTION-LIST-NAME>

Specifies the action list name to have its entries resequenced.

<STARTING-SEQUENCE-NUMBER>

Specifies the starting sequence number. Range: 1-4294967295

<INCREMENT>

Specifies the increment of the resequencing. Range: 1-4294967295

Authority

Administrators or local user group members with execution rights for this command.

Examples

The following command shows how a PBR action list is resequenced. In the following example, an action list named test4 is resequenced so that instead of its entries starting at 5 and being numbered sequentially, its entries start now at 1 and they are numbered in increments of 10:

```
switch(config)# show pbr-action-list test4
```

Sequence	Name Type	Address/Interface

	test4	
5	nexthop	2000.abcd::cccc.ddd
6	nexthop	1000.abcd::1234.5678
7	interface	null

```
switch(config)# pbr-action-list test4 resequence 1 10
```

Sequence	Name Type	Address/Interface

	test4	
1	nexthop	2000.abcd::cccc.ddd
11	nexthop	1000.abcd::1234.5678
21	interface	null

pbr-action-list reset

Syntax

```
pbr-action-list <ACTION-LIST-NAME> reset
```

Description

Resets a specified PBR action list to its last successful configuration.

Command context

```
config
```

Parameters

<ACTION-LIST-NAME>

Specifies the action list name to be reset.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# pbr-action-list test reset
```

policy

Syntax

```
policy <POLICY-NAME>
```

```
[<SEQUENCE-NUMBER>]
  class {ip|ipv6|mac} <CLASS-NAME>
    action {<REMARK-ACTIONS> | <POLICE-ACTIONS> | <OTHER-ACTIONS>}
    [{<REMARK-ACTIONS> | <POLICE-ACTIONS> | <OTHER-ACTIONS>}]

[<SEQUENCE-NUMBER>]
  comment ...
```

```
no policy <POLICY-NAME>
```

Description

Creates, modifies, or deletes a classifier policy. A policy contains one or more policy entries ordered and prioritized by sequence numbers. Each entry has an IPv4/IPv6/MAC class and one or more policy actions associated with it. An applied policy processes a packet sequentially against policy entries in the list until the last entry in the list has been evaluated or the packet matches an entry. If a match occurs the related entry, actions are taken.

The `no` form of this command is used to delete a policy or an individual policy entry.

Command context

```
config
```

The `policy` command takes you into the `config-policy` context where you enter the policy entries.

Parameters

<POLICY-NAME>

Specifies the name of the policy.

<SEQUENCE-NUMBER>

Specifies a sequence number for the policy entry. Optional. Range: 1 to 4294967295.

comment

Stores the remaining entered text as a policy entry comment.

class {ip|ipv6|mac} <CLASS-NAME>

Specifies a type of class, `ip` for IPv4, `ipv6` for IPv6 and `mac` for a MAC policy. And specifies a class name.

<REMARK-ACTIONS>

Remark actions can be any of the following options: {pbr <ACTION-LIST> | pcp <PRIORITY> | ip-precedence <IP-PRECEDENCE-VALUE> | dscp <DSCP-VALUE> | local-priority <LOCAL-PRIORITY-VALUE>} where:

pbr <ACTION-LIST>

Specifies the PBR action list to be used.

pcp <PCP-VALUE>

Specifies Priority Code Point (PCP) value. Range: 0 to 7.

ip-precedence <IP-PRECEDENCE-VALUE>

Specifies the numeric IP precedence value. Range: 0 to 7.

dscp <DSCP-VALUE>

Specifies a Differentiated Services Code Point (DSCP) value. Enter either a numeric value (0 to 63) or a keyword as follows:

- AF11 - DSCP 10 (Assured Forwarding Class 1, low drop probability)
- AF12 - DSCP 12 (Assured Forwarding Class 1, medium drop probability)
- AF13 - DSCP 14 (Assured Forwarding Class 1, high drop probability)
- AF21 - DSCP 18 (Assured Forwarding Class 2, low drop probability)

- AF22 - DSCP 20 (Assured Forwarding Class 2, medium drop probability)
- AF23 - DSCP 22 (Assured Forwarding Class 2, high drop probability)
- AF31 - DSCP 26 (Assured Forwarding Class 3, low drop probability)
- AF32 - DSCP 28 (Assured Forwarding Class 3, medium drop probability)
- AF33 - DSCP 30 (Assured Forwarding Class 3, high drop probability)
- AF41 - DSCP 34 (Assured Forwarding Class 4, low drop probability)
- AF42 - DSCP 36 (Assured Forwarding Class 4, medium drop probability)
- AF43 - DSCP 38 (Assured Forwarding Class 4, high drop probability)
- CS0 - DSCP 0 (Class Selector 0: Default)
- CS1 - DSCP 8 (Class Selector 1: Scavenger)
- CS2 - DSCP 16 (Class Selector 2: OAM)
- CS3 - DSCP 24 (Class Selector 3: Signaling)
- CS4 - DSCP 32 (Class Selector 4: Real time)
- CS5 - DSCP 40 (Class Selector 5: Broadcast video)
- CS6 - DSCP 48 (Class Selector 6: Network control)
- CS7 - DSCP 56 (Class Selector 7)
- EF - DSCP 46 (Expedited Forwarding)

local-priority *<LOCAL-PRIORITY-VALUE>*

Specifies a local priority value. Range: 0 to 7.

<POLICE-ACTIONS>

Police actions can be the following {cir *<RATE-BPS>* cbs *<BYTES>* exceed} where:

cir *<RATE-BPS>*

Specifies a Committed Information Rate value in Kilobits per second. Range: 1 to 4294967295.

cbs *<BYTES>*

Specifies a Committed Burst Size value in bytes. Range: 1 to 4294967295.

exceed

Specifies action to take on packets that exceed the rate limit.

<OTHER-ACTIONS>

Other actions can be the following:

drop

Specifies drop traffic.

Authority

Administrators or local user group members with execution rights for this command.

Restrictions

MAC classes are not applicable to policies containing PBR actions. Applying such policies to an interface are blocked.

Usage

- For Policy Based Routing, the policy action keyword is `pbr` which itself takes the name of a PBR action list as a parameter.
- A policy entry that contains a PBR action can contain other action types as well.
- An applied policy processes a packet sequentially against policy entries in the list until the last policy entry in the list has been evaluated or the packet matches an entry.
- Entering an existing `<POLICY-NAME>` value will cause the existing policy to be modified, with any new `<SEQUENCE-NUMBER>` value creating an additional policy entry, and any existing `<SEQUENCE-NUMBER>` value replacing the existing policy entry with the same sequence number.
- If no sequence number is specified, a new policy entry is appended to the end of the entry list with a sequence number equal to the highest policy entry currently in the list plus 10.

Examples

Create a policy with two PBR actions:

```
switch(config)# policy pbr_policy
switch (config-policy)# 10 class ip v4_class action pbr action_list1
switch (config-policy)# 20 class ipv6 v6_class action pbr action_list2
switch (config-policy)# exit
```

show pbr

Syntax

```
show pbr {interface <INTERFACE-NAME>|vrf <VRF-NAME>|summary}
```

Description

Shows a detailed view of Policy Based Routing (PBR) in the system.

Command context

Operator (>) or Manager (#)

Parameters

<VRF-NAME>

Specifies name of a VRF.

<INTERFACE-NAME>

Specifies an interface. Format: member/slot/port.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing PBR summary information when there is no active next-hop in the system:

```
switch# show pbr summary
VRF      Port    Policy    PBR      Seq  Type      Nexthop
-----
No active PBR nexthop found
-----
```

Showing PBR summary information when there are active next-hops in the system:

```
switch# show pbr summary
VRF      Port    Policy    PBR      Seq  Type      Nexthop
-----
default  1/1/1  policy_1  pbr_1    10   nexthop    1.1.1.1 (active)
          1/1/2  policy_2  pbr_2    20   nexthop    5.5.5.5 (active)
-----
```

show pbr-action-list

Syntax

```
show pbr-action-list [<ACTION-LIST-NAME>] [commands] [configuration] [vsx-peer]
```

Description

Shows the current PBR action list configuration. Action list entries are displayed in ascending order of their sequence number.

Command context

config

Parameters

<ACTION-LIST-NAME>

Specifies the PBR action list name.

commands

Formats output as CLI commands.

configuration

Displays user-specified configuration.

vsx-peer

Displays VSX peer switch information.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Restrictions

If an action list entry is modified to an invalid value (for example through the REST interface), this command will indicate a mismatch for that action entry when run. In this event, use the `pbr-action-list <NAME> reset` command to restore it to the previous valid value.

Usage

- This command does not indicate whether the action list is configured in a policy or applied to an interface. Use the `show pbr` command for PBR status involving action lists.
- A single action list is shown by specifying its name or you can show all action lists by omitting a name argument.
- Using the additional `commands` keyword, you can change the tabulated output to a configuration style output for single or all list display.

Examples

Create two PBR action lists then run `show pbr-action-list` to display all configured action lists in the default configuration mode:

```
switch(config)# pbr-action-list v4_pbr
switch(config-pbr-action-list-v4_pbr)# 1 nexthop 1.1.1.1
switch(config-pbr-action-list-v4_pbr)# 5 default-nexthop 2.2.2.2
switch(config-pbr-action-list-v4_pbr)# 10 interface null
switch(config-pbr-action-list-v4_pbr)# exit
switch(config)#
switch(config)# pbr-action-list v6_pbr
switch(config-pbr-action-list-v6_pbr)# 20 nexthop 2000:abcd::cccc:dddd
switch(config-pbr-action-list-v6_pbr)# 40 default-nexthop 1000:abcd::1234:5678
switch(config-pbr-action-list-v6_pbr)# 60 interface null
switch(config-pbr-action-list-v6_pbr)# exit
switch#
```

```
switch# show pbr-action-list
```

	Name	
	Additional PBR-Action-List Parameters	
Sequence	Type	Nexthop

	v4_pbr	
1	nexthop	1.1.1.1
5	default-nexthop	2.2.2.2
10	interface	null
	v6_pbr	
20	nexthop	2000:abcd::cccc:dddd
40	default-nexthop	1000:abcd::1234:5678
60	interface	null

show running-config current-context

Syntax

```
show running-config current-context
```

Description

Displays the configuration of the PBR action list in the current configuration context, in commands mode.

Parameters

running-config

Shows configuration currently running on switch.

current-context

Limits display to current config context only, in commands mode.

Authority

Administrators or local user group members with execution rights for this command.

Usage

Useful for reexamining entries previously entered into the action list after its entries have scrolled off the terminal due to other output or upon reentering the context of an existing action list.

Examples

Creating two PBR action lists and running `show running-configuration curenent-context` to display the action list configuration in commands mode:

```
switch(config)# pbr-action-list v4_pbr
switch(config-pbr-action-list-v4_pbr)# 1 nexthop 1.1.1.1
switch(config-pbr-action-list-v4_pbr)# 5 default-nexthop 2.2.2.2
switch(config-pbr-action-list-v4_pbr)# 10 interface null
switch(config-pbr-action-list-v4_pbr)# exit
switch(config)#
switch(config)# pbr-action-list v6_pbr
switch(config-pbr-action-list-v6_pbr)# 20 nexthop 2000:abcd::cccc:dddd
switch(config-pbr-action-list-v6_pbr)# 40 default-nexthop 1000:abcd::1234:5678
switch(config-pbr-action-list-v6_pbr)# 60 interface null
switch(config-pbr-action-list-v6_pbr)#
switch(config-pbr-action-list-v6_pbr)# show running-config current-context
pbr-action-list v6_pbr
    20 nexthop 2000:abcd::cccc:dddd
    40 default-nexthop 1000:abcd::1234:5678
    60 interface null
```

Switching context back to the first action list and running the same command:

```
switch(config-pbr-action-list-v6_pbr)# pbr-action-list v4_pbr
switch(config-pbr-action-list-v4_pbr)#
switch(config-pbr-action-list-v4_pbr)# show running-config current-context
pbr-action-list v4_pbr
    1 nexthop 1.1.1.1
    5 default-nexthop 2.2.2.2
    10 interface null
```

Removing action list entry number 5 and running the command again:

```
switch(config-pbr-action-list-v4_pbr)# no 5
switch(config-pbr-action-list-v4_pbr)# show running-config current-context
pbr-action-list v4_pbr
    1 nexthop 1.1.1.1
    10 interface null
```

IP Directed Broadcast Overview

IP Directed Broadcast is a feature by which remote administration tasks such as backups and wake-on-LAN (WOL) application can be achieved by sending directed broadcast packets for hosts/servers residing on a different subnet. The datagram is routed by normal mechanisms until it reaches a gateway attached to the destination hardware network, at which point it is broadcast. This class of broadcasting is also known as 'directed broadcasting'. IP Directed Broadcast packets are Layer 3 subnet broadcasts.

For example, for the broadcast IP address 192.168.1.255, the corresponding egress interface Subnet IP would be 192.168.1.x/24. Classless subnetting (CIDR) is also supported. For example, for the broadcast IP address 10.10.15.255, the broadcast address for the corresponding interface IP would be 10.10.15.x/21.

IP Directed Broadcast is supported on Route Only Port (ROP), Switched Virtual Interface (SVI), and Layer 3 Link Aggregation Group (L3LAG) interfaces but is disabled by default. The feature is supported for the associated interface addresses for both the primary and secondary addresses. IP directed broadcast can be enabled on all the configured L3 interfaces. It is only supported for IPv4 addresses.

IP Directed Broadcast configuration example

The following are sample topology diagrams for an IP Directed Broadcast configuration. Figure 1 shows when the egress interface is an ROP, while figure 2 shows the diagram when the egress interface is an SVI.

Figure 1: ROP: IP Directed Broadcast

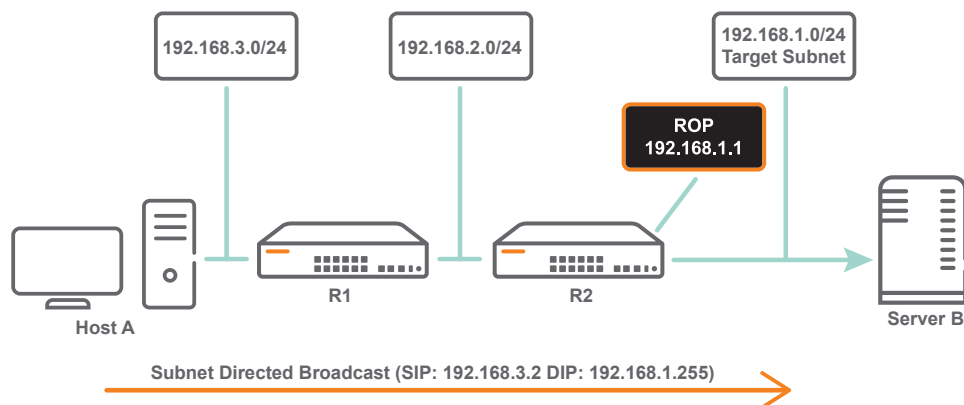
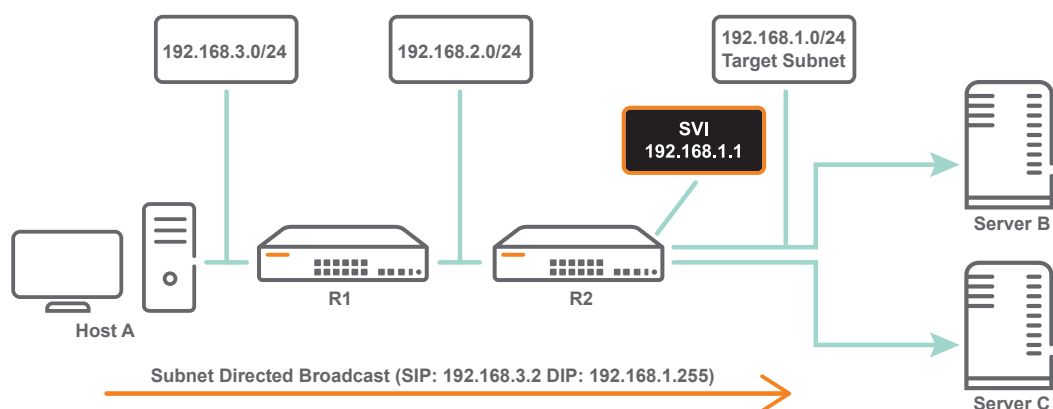


Figure 2: SVI: IP Directed Broadcast



Intermediate routers forward IP Directed Broadcast packets as Unicast. The IP directed broadcast packet is broadcast or flood in the target subnet (DA MAC: All 0xFFs) only after the last hop router.

Host A in subnet 192.168.3.0/24 wants to inject a IP Directed Broadcast (192.168.1.255) packet into Target Subnet 192.168.1.0/24. Router R1 forwards the IP Datagram with DIP 192.168.1.255 as a regular Unicast Datagram. Router R2 then floods the IP Datagram over egress ROP or SVI with Destination MAC as all 0xFFs.

At Ingress, Port Based ACLs (PACL) and VLAN Based ACLs (VACL) can be used to restrict/allow IP Directed Broadcast traffic. Existing Port based ACLs (PACL) can be used to allow or disallow certain IP Directed Broadcast Traffic.

An ACL can be configured using the `access-list ip <ACL-NAME>` command and then applied using the `apply access-list ip <ACL-NAME>` command as shown in the following output.

```
switch(config)# access-list ip ipdbacl

switch(config)# interface 1/1/1
switch(config-if)# apply access-list ipdbacl
    in   Inbound (ingress) traffic
    out  Outbound (egress) traffic

switch(config-if)# int lag 10
```

```
switch(config-lag-if)# apply access-list ipdbacl
  in   Inbound (ingress) traffic
  out  Outbound (egress) traffic
```

The following is an example of the `show running-config` command on an ROP interface.

```
switch(config)# interface 1/1/1
  no shutdown
  ip address 192.168.1.1/24
  ip directed-broadcast
```

The following is an example of the `show running-config` command on an SVI interface.

```
switch(config)# vlan 10
interface vlan10
  no shutdown
  ip address 192.168.1.1/24
  ip directed-broadcast
```

The following is an example of the `show running-config` command on an L3LAG interface.

```
switch(config)# interface lag 3
  no shutdown
  ip address 192.168.1.1/24
  ip directed-broadcast
```



NOTE: Note: Currently egress ACL is supported only on ROP and LAG interfaces, and not on an SVI interface.

IP Directed Broadcast commands

copy support-file feature

Syntax

```
copy support-file feature 13
```

Description

Captures support logs to debug any IP Directed Broadcast issues.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Capturing the support logs into a local file:

```
switch# copy support-file feature 13 sftp
```

ip directed-broadcast

Syntax

```
ip directed-broadcast
no ip directed-broadcast
```

Description

Turns on IP Directed Broadcast for the specified interface. The `no` form of this command turns it off. This command is disabled by default.

Command context

```
config-if
```

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Enabling and disabling IP Directed Broadcast on an L3 physical interface:

```
switch(config-if)# ip directed-broadcast
switch(config-if)# no ip directed-broadcast
```

Enabling and disabling IP Directed Broadcast on a VLAN interface:

```
switch(config-if-vlan)# ip directed-broadcast
switch(config-if-vlan)# no ip directed-broadcast
```

show ip interface

Syntax

```
show ip interface <INTERFACE-NAME>
```

Description

Displays the status of IP Directed Broadcast on the specified interface along with other interface related attributes.

Command context

Operator (>) or Manager (#)

Parameters

<INTERFACE-NAME>

Specifies the interface to use as a source for displaying the status of the IP Directed Broadcast.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying the IP Directed Broadcast status on the specified interface:

```
switch# show ip interface vlan30
```

```
Interface vlan30 is up
Admin state is up
Hardware: Ethernet, MAC Address: 94:f1:28:21:63:00
IP MTU 1500
```

```
IP Directed Broadcast is Enabled
IPv4 address 192.168.3.1/24
L3 Counters: Rx Disabled, Tx Disabled
```

show ip directed-broadcast

Syntax

```
show ip directed-broadcast
```

Description

Displays the summary of the interfaces on which IP Directed Broadcast is enabled.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

On the 6400 Switch Series, interface identification differs.

Displaying the summary of the interfaces on which IP Directed Broadcast is enabled:

```
switch# show ip directed-broadcast
```

```
IPv4 Directed Broadcast Configuration
```

Interface	Status
1/1/1	Enabled
vlan10	Enabled
vlan30	Enabled

Displaying IP Directed Broadcast Host entries installed in Neighbor cache:

```
switch# show arp state permanent
```

IPv4 Address	MAC	Port	Physical Port	State
52.1.1.255	FF:FF:FF:FF:FF:FF	1/1/1	1/1/1	permanent
40.0.0.255	FF:FF:FF:FF:FF:FF	vlan20	vlan20	permanent
Total Number Of ARP Entries Listed- 2.				

IP Neighbor Flood Overview



NOTE: IP Neighbor Flood is only supported on the Aruba 8320, 8325 and 8360 Switch Series.

When a port goes down, L3 traffic to a neighbor will get dropped until the MAC address is learned on the new port. Source MAC learning could take a while, and this learning could be dependent on traffic being successfully delivered to the client.

An example is when wireless clients are connected through access points balanced across several wireless controllers connected to Halon switches. Traffic to clients is dropped when the port connecting the controller and switch goes down.

The port can go down when the wireless controller goes down and reboots. Access points are adopted by other controllers on the network to avoid traffic drop for the wireless clients, but the back-end data plane requires handling.

During such failover conditions in scaled setups, MAC learning could take a while, and significant traffic loss can occur.

The solution is to flood traffic on the VLAN so that traffic reaches the hosts.

Once neighbors are relearned over a new port on the same VLAN, the switch will stop flooding.

IP Neighbor Flood is supported on SVIs. It is disabled by default.

IP Neighbor Flood commands

`ip neighbor-flood`

Syntax

```
ip neighbor-flood
```

Description

Enables VLAN flooding for the specified VLAN interface when a neighbor link goes down.

The `no` form of this command disables VLAN flooding for the specified VLAN interface.

Command context

```
config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling IP Neighbor Flood on a VLAN interface:


```
switch(config)# interface vlan 3  
switch(config-if-vlan)# ip neighbor-flood
```

Disabling IP Neighbor Flood on a VLAN interface.

```
switch(config)# interface vlan 3  
switch(config-if-vlan)# no ip neighbor-flood
```

show ip interface

Syntax

```
show ip interface <IFNAME>
```

Description

Displays the status of IP Neighbor Flood on the specified interface along with other interface-related attributes.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Parameters

<IFNAME>

Specifies the interface name (for example, `vlan30`). Optional.

Examples

```
switch# show ip interface vlan30  
  
Interface vlan30 is up  
Admin state is up  
Hardware: Ethernet, MAC Address: 94:f1:28:21:63:00  
IP MTU 1500  
IP Neighbor Flood is Enabled  
IPv4 address 192.168.3.1/24  
L3 Counters: Rx Disabled, Tx Disabled
```

show ip neighbor-flood

Syntax

```
show ip neighbor-flood
```

Description

Displays the interfaces on which IP Neighbor Flood is enabled.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

```
switch# show ip neighbor-flood
```

IP Neighbor Flood Configuration

Interface	Status
-----	-----
vlan10	Enabled
vlan30	Enabled

show running-config

Syntax

```
show running-config
```

Description

Displays the current running configuration.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command.
Operators can execute this command from the operator context (>) only.

Examples

```
switch# show running-config
interface vlan10
    ip neighbor-flood
interface vlan30
    ip neighbor-flood
```

Overview

This chapter provides details for configuring and verifying the functions of a key chain. Key chain configurations are spread across `config` context, `keychain` context, and `keychain-key` context.

A key chain provides seamless authentication-key rollover. When authentication method is enabled, using key chain allows OSPF to overcome static key configuration to change the key periodically. Using key chain also reduces risks of keys being guessed by configuring rotating keys.



NOTE: A key chain must be used by an application like OSPF that communicates by using the keys with its peers.

Key chain commands

accept-lifetime

Syntax

```
accept-lifetime [start-time <time> <month>/<day>/<year>] {duration {<seconds> | infinite} | end-time <time> <month>/<day>/<year>}
```

Description

Configures the duration for which the key is valid for receiving packets.

The `no` form of this command configures the key packet receiving duration to the default value of an infinite time.

Command context

`config-keychain-key`

Parameters

start-time

Time at which the key chain lifetime starts. Required. Format: HH:MM:SS

end-time

Time at which the key chain lifetime expires. Required. Format: HH:MM:SS

day

Day of the month. Required. Range: 1-31.

month

Month of the year. Required.

year

Year. Required. Range: 2020-2050

duration

Time in seconds. Optional. Range: 1-2147483646.

infinite

Specifies infinite time for the key. Optional.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Configuring the duration for which the key is valid for receiving packets:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# accept-lifetime start-time 10:10:10 10/25/2020 end-time 10:10:10 11/25/2020
switch(config-keychain-key)# accept-lifetime start-time 10:10:10 10/25/2020 duration 1000
switch(config-keychain-key)# accept-lifetime start-time 10:10:10 10/25/2020 duration infinite
switch(config-keychain-key)# accept-lifetime end-time 10:10:10 11/25/2020
switch(config-keychain-key)# accept-lifetime duration 1000
switch(config-keychain-key)# accept-lifetime duration infinite
```

Configuring the key packet receiving duration to the default value of an infinite time:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# no accept-lifetime
```

key

Syntax

key <key-id>

Description

Creates the key for a key chain and enters the key chain key context. A maximum of 64 keys can be configured per key chain.

The **no** form of this command deletes the key from the key chain.

Command context

config-keychain

Parameters

key-id

ID of the key. Required. Range: 1-255.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Creating a key for a key chain:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
```

Deleting a key from a key chain:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# no key 1
```

keychain

Syntax

```
keychain <keychain-name>
```

Description

Creates the key chain and enters the key chain context. A maximum of 64 key chains can be configured in the system.

The `no` form of this command removes the key chain if it is not used by any subscribers.

Command context

```
config
```

Parameters

keychain-name

Name of the key chain. Required.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Creating a key chain:

```
switch# configure terminal
switch(config)# keychain ospf_keys
```

Removing a key chain:

```
switch# configure terminal
switch(config)# keychain ospf_keys
```

key-string

Syntax

```
key-string {ciphertext | plaintext} <key>
```

Description

Sets the password used for the key. The password is always stored in encrypted format in the OVSDb. The key will be considered invalid if the password is not specified for the key.

The `no` form of this command deletes the password used for the key.

Command context

`config-keychain-key`

Parameters

ciphertext

Password in encrypted format. Required.

plaintext

Password in simple-text format. Required.

key

Password for key authentication. Required.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Setting the password for the key:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string plaintext key
<cr>
switch(config-keychain-key)# key-string ciphertext ASDFkjdkj
<cr>
```

Deleting the password from the key:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# no key-string
<cr>
```

send-lifetime

Syntax

```
send-lifetime [start-time <time> <month>/<day>/<year>] {duration {<seconds> | infinite} | end-time <time> <month>/<day>/<year>}
```

Description

Configures the duration for which the key is valid for sending packets.

The `no` form of this command configures the key packet sending duration to the default value of an infinite time.

Command context

`config-keychain-key`

Parameters

start-time

Time at which the key chain lifetime starts. Required. Format: HH:MM:SS

end-time

Time at which the key chain lifetime expires. Required. Format: HH:MM:SS

day

Day of the month. Required. Range: 1-31.

month

Month of the year. Required.

year

Year. Required. Range: 2020-2050

duration

Time in seconds. Optional. Range: 1-2147483646.

infinite

Specifies infinite time for the key. Optional.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Configuring the duration for which the key is valid for sending packets:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# send-lifetime start-time 10:10:10 10/25/2020 end-time 10:10:10 11/25/2020
switch(config-keychain-key)# send-lifetime start-time 10:10:10 10/25/2020 duration 1000
switch(config-keychain-key)# send-lifetime start-time 10:10:10 10/25/2020 duration infinite
switch(config-keychain-key)# send-lifetime end-time 10:10:10 11/25/2020
switch(config-keychain-key)# send-lifetime duration 1000
switch(config-keychain-key)# send-lifetime duration infinite
```

Configuring the key packet sending duration to the default value of an infinite time:

```
switch# configure terminal
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# no send-lifetime
```

show capacities keychain

Syntax

```
show capacities keychain
```

Description

Shows the maximum number of key chains and keys configurable in a key chain.

Command context

Operator (>)

Parameters

None

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch# show capacities keychain
```

```
System Capacities: Filter Keychain
Capacities Name                                     Value
-----
Maximum number of keychains supported in the system  64
Maximum number of Keys supported in a single Keychain 64
```

show keychain

Syntax

```
show keychain [<keychain-name>]
```

Description

Shows information about configured and active keys of a named key chain or (if `keychain-name` is not specified) all configured key chains.

Command context

Operator (>)

Parameters

keychain-name

Name of the key chain. Optional.

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch# show keychain
```

```
Keychain Name : ospf_keys
```

```
  Number of Keys      : 2
```

```
  Active Send Key ID  : 7
```

```
  Active Recv Key IDs : 7, 200
```

```
Key ID                : 7
```

```
  Key string           : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
```

```
  Send Key Validity    : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
```

```
  Recv Key Validity    : 00:00:01 10/1/2020 to infinite
```

```
Key ID                : 200
```

```
  Key string           : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
```

```
  Send Key Validity    : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
```

```
  Recv Key Validity    : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
```

```
Keychain Name : bgp_keys
```

```
  Number of Keys      : 2
```

```
  Active Send Key ID  : 7
```


Active Recv Key IDs : 7

```
Key ID          : 7
Key string      : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
Send Key Validity : 00:00:01 10/26/2020 to 23:59:01 10/1/2021
Recv Key Validity : 00:00:01 10/22/2020 to infinite
Key ID          : 8
Key string      : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
Send Key Validity : 00:00:01 10/1/2021 to 23:59:01 10/1/2021
Recv Key Validity : 00:00:01 10/1/2021 to 23:59:01 10/1/2021
```

Keychain Name : ospf_keys
Number of Keys : 2
Active Send Key ID : 7
Active Recv Key IDs : 7, 200

```
Key ID          : 7
Key string      : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
Send Key Validity : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
Recv Key Validity : 00:00:01 10/1/2020 to infinite
Key ID          : 200
Key string      : AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
Send Key Validity : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
Recv Key Validity : 00:00:01 10/1/2020 to 23:59:01 10/1/2021
```

show running-config keychain

Syntax

show runnning-config keychain

Description

Shows the configurations for key chain protocol.

Command context

Operator (>)

Parameters

None

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch# show running-config keychain
keychain ospf_keys
key 1
key-string ciphertext AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
accept-lifetime start-time 10:10:10 10/25/2020 end-time 10:10:10 11/25/2020
send-lifetime start-time 10:10:10 10/25/2020 end-time 10:10:10 11/25/2020
key 45
key-string ciphertext AQBapZ10HiO9W3JwRqnjtLfbV73BPLS1S6TGVg+Lzl7N4e5eBAAAAPWaPBE=
accept-lifetime start-time 10:10:10 10/25/2020 end-time 10:10:10 11/25/2020
```

```
key 33  
switch#
```

Overview



NOTE: IP Client Tracker is only supported on the Aruba 6x00 Switch Series.

The client IP address tracking feature will learn and update the IP addresses of the access devices and clients connected to the switch. It can track addresses of directly connected clients, as well as clients connected to a downstream device such as a wireless access point.

IP Client Tracker commands

client track ip

Syntax

```
client track ip
```

Description

Enables client IP address tracking on the switch. The default is disabled on global and VLAN levels.

Admin users can enable client IP address tracking at the VLAN level.



NOTE: Tracking enabling will take effect only if the client IP address tracking is enabled at system and VLAN level.

The `no` form of the command disables client IP address tracking. If tracking is disabled at switch level, it will be stopped even if it is enabled at VLAN or port level.

Command context

```
config
```

Parameters

None

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Enable client IP address tracking at switch level:

```
switch(config)# client track ip
```

Enable client IP address tracking on VLAN 100:

```
switch(config)# vlan 100
switch(config-vlan-100)# client track ip
```

Enable client IP address tracking on VLANs 10 to 100:

```
switch(config)# vlan 10-100
```

```
switch(config-vlan-<10-100>)# client track ip
```

client track ip { enable | disable | auto }

Syntax

```
client track ip { enable | disable | auto }
```

Description

Enables client IP address tracking on the specified set of interfaces. Tracking will take effect only if client IP address tracking is enabled at both the system level and for the VLAN to which the port belongs. Default: auto.

The `no` form of the command disables client IP address tracking on the specified set of interfaces.

Command context

```
config-if
```

Parameters

enable

Specifies that all client IP addresses will be tracked in the port.

disable

Specifies that client IP addresses will not be tracked in the port.

auto

Specifies the following:

- For LLDP devices: Only the specified client IP address will be tracked in the port and other client IP addresses will not be tracked.
- For non-LLDP devices: All client IP addresses will be tracked in the port.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Enable client IP address tracking on interface 1/1/1:

```
switch(config)# interface 1/1/1
```

```
switch(config-if)# client track ip enable
```

Enable client IP address tracking on interfaces 1/1/1 to 1/1/5:

```
switch(config)# interface 1/1/1-1/1/5
```

```
switch(config-if-<1/1/1-1/1/5>)# client track ip enable
```

client track ip client-limit

Syntax

```
client track ip client-limit <CLIENT-LIMIT>
```

Description

Configures the maximum number of clients to be tracked on the specified set of interfaces.

The `no` form of the command resets the client limit to the default value. Default values vary according to switch model:

- 6300: 2048
- 6400: 4096

Command context

`config-if`

Parameters

CLIENT-LIMIT

Specifies the maximum number of clients tracked on a port. Required. Range: 1-2048 (6300) 1-4096 (6400). Default: 2048 (6300) 4096 (6400).

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Configure the maximum number of clients to be tracked on interface 1/1/5:

```
switch(config)# interface 1/1/5
switch(config-if)# client track ip client-limit 32
```

`client track ip update-interval`

Syntax

```
client track ip update-interval <INTERVAL>
```

Description

Configures how often client IP addresses are updated.

The `no` form of the command resets the update interval to the default of 1800 seconds.

Command context

`config-if`

Parameters

INTERVAL

Specifies the update interval in seconds. Required. Range: 60-28000. Default: 1800.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Configure the update interval for an interface:

```
switch(config)# interface 1/1/1
switch(config-if)# client track ip update-interval 600
```

client track ip update-method probe

Syntax

```
client track ip update-method probe
```

Description

Enables probing the client to update the IP address.

The probe is sent to all clients on the tracking list that have an IP address in the following scenarios:

1. IP packets are not received from the clients during the IP address update cycle.
2. There is no IP packet from a learned IP address. In this case, a probe will be sent for the IP address to confirm if it is still owned by that client.

The `no` form of the command disables probing.

Command context

```
config
```

Parameters

None

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Disable probing to update the client IP address:

```
switch(config)# no client track ip update-method probe
```

show capacities

Syntax

```
show capacities
```

Description

Shows the capacities configured on the switch.

Command context

Operator (>)

Parameters

None

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch# show capacities
```

```
System Capacities:
Capacities Name                                     Value
-----
Maximum number of Access Control Entries configurable in a system 14336
Maximum number of Access Control Lists configurable in a system    1024
Maximum number of class entries configurable in a system           1024
Maximum number of classes configurable in a system                 512
Maximum number of entries in an Access Control List                1024
Maximum number of entries in a class                              1024
Maximum number of entries in a policy                              1024
Maximum number of classifier policies configurable in a system      512
Maximum number of policy entries configurable in a system           1024
Maximum number of clients supported for tracking the IP address in the system 128
switch# show capacities client-track-ip-client-limit
```

```
System Capacities: Filter Client Track IP Client Limit
Capacities Name                                     Value
-----
Maximum number of clients supported for tracking the IP address in the system
```

```
show client ip { count | port | vlan }
```

Syntax

```
show client ip { count | port | vlan }
```

Description

Shows number of client IP addresses or information about client IP addresses tracked on ports and VLANs.

Command context

Operator (>)

Parameters

count

Displays number of clients tracked.

port

Displays client IP addresses tracked on the ports.

vlan

Displays client IP addresses tracked on the VLANs.

Authority

Administrators or local user group members with execution rights for this command.

Overview

Routing Information protocol (RIP, RIPv2, RIPv6) is a distance-vector routing protocol that uses hop count as a routing metric. It is useful in small networks. RIP can be configured on L3 ports, LAG, VLAN and Loopback interfaces. All configurations work in the interface context and require the associated interface to be routing interface.

- **RIP/RIPv2:** IPv4 implementation of Routing Information protocol defined in RFC 2453.
- **RIPv6:** IPv6 implementation of Routing Information protocol defined in RFC 2080.

RIP characteristics:

- Prevents routing loops by adding a limit to the number of hops allowed in a path from source to destination.
- Allows a maximum of 15 hops which limits the size of the network that RIP can support.
- Uses broadcast UDP data packets to exchange routing information.
- RIPv2 is a classless protocol that supports variable-length subnet mask (VLSM), classless inter-domain routing (CIDR) and route summarization.
- RIPv6 uses ipsec for message authentication.

RIP limitations:

- RIPv1 is not supported.
- Only poison reverse is supported for loop prevention.

RIPv2 (IPv4) commands

Configuration commands

`router rip`

Syntax

```
router rip <PROCESS-ID> [vrf <VRF-NAME>]
```

```
no router rip <PROCESS-ID> [vrf <VRF-NAME>]
```

Description

Creates RIP process if not already created and enters the `router rip <PROCESS-ID>` context for the VRF mentioned. If no VRF is mentioned, a default is used. Only one RIP process is allowed per VRF.

The `no` form of this command deletes the RIP instance for the VRF. If no VRF is mentioned the default is deleted.

Command context

config

Parameters

<PROCESS-ID>

Specifies name of the RIP process ID. Range: 1-63.

vrf <VRF-NAME>

Specifies VRF name.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating RIP process and naming the VRF:

```
switch(config)# router rip 2 vrf red
```

Deleting RIP process:

```
switch(config)# no router rip 2 vrf red
```

Interface commands

ip rip

Syntax

```
ip rip <PROCESS-ID> {all-ip | ip-address}
```

```
no ip rip <PROCESS-ID> {all-ip | ip-address}
```

Description

Enables RIP process on an interface.

The **no** form of this command deletes the RIP process from an interface.

Command context

config-if

Parameters

ip rip <PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

all-ip

Specifies RIP for all IP addresses configured on the interface.

ip-address

Specifies IP address for RIP on the interface.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.
- If `ip rip 1 all-ip` is configured and a new IP address is added to the interface, RIP configurations will not be applicable for the newly added IP address.

Examples

Configuring RIP for all IP addresses configured on the interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip 1 all-ip
```

Deleting RIP for all IP addresses configured on the interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# no ip rip 1 all-ip
```

`ip rip all-ip enable`

Syntax

```
ip rip all-ip enable
```

```
no ip rip all-ip enable
```

Description

Enables RIP process for all RIP enabled IP addresses configured on interface.

The `no` form of this command disables RIP process on the interface.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Enabling RIP process for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip all-ip enable
```

Disabling RIP process on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# no ip rip all-ip enable
```

ip rip all-ip disable

Syntax

```
ip rip all-ip disable
```

```
no ip rip all-ip disable
```

Description

Disables RIP process for all RIP enabled IP addresses configured on the interface.

The `no` form of this command enables RIP process for all RIP enabled IP addresses configured on the interface.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Disabling RIP process for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip all-ip enable
```

Enabling RIP process for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# no ip rip all-ip enable
```

ip rip all-ip send disable

Syntax

```
ip rip all-ip send disable
```

```
no ip rip all-ip send disable
```

Description

Disables interface from sending RIP packets for all RIP enabled IP addresses.

The `no` form of this command enables interface to send RIP packets for all RIP enabled IP addresses.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to send RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Disabling interface from sending RIP packets for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip all-ip send disable
```

Enabling interface to send RIP packets for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# no ip rip all-ip send disable
```

ip rip all-ip receive disable

Syntax

ip rip all-ip receive disable

no ip rip all-ip receive disable

Description

Disables interface from receiving RIP packets for all enabled IP addresses.

The **no** form of this command enables interface to receive RIP packets for all RIP enabled IP addresses.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Disabling interface from receiving RIP packets for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip all-ip receive disable
```

Enabling interface to receive RIP packets for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# no ip rip all-ip receive disable
```

Routing commands

enable

Syntax

```
enable
```

```
no enable
```

Description

Enables RIP process if disabled. By default RIP process is enabled.

The `no` form of this command disables the RIP process.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling RIP process when disabled:

```
switch(config)# router rip 1  
switch(config-rip-1)# enable
```

Disabling RIP process when enabled:

```
switch(config)# router rip 1  
switch(config-rip-1)# no enable
```

disable

Syntax

```
disable
```

```
no disable
```

Description

Disables RIP process.

The `no` form of this command enables the RIP process.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling RIP process:

```
switch(config)# router rip 1  
switch(config-rip-1)# disable
```

Enabling RIP process:

```
switch(config)# router rip 1  
switch(config-rip-1)# no disable
```

distance

Syntax

```
distance <DISTANCE>
```

```
no distance
```

Description

Configures administrative distance for RIP. Administrative distance is used as criteria to select the best route when multiple protocols have the same route.

The `no` form of this command sets the RIP administrative distance to the default. Default: 120.

Command context

```
config
```

Parameters

<DISTANCE>

Specifies RIP administrative distance. Range: 1 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring administrative distance for RIP:

```
switch(config)# router rip 1  
switch(config-rip-1)# distance 100
```

Setting administrative distance for RIP to default values:

```
switch(config)# router rip 1  
switch(config-rip-1)# no distance
```

maximum-paths

Syntax

```
maximum-paths <MAX-VALUE>
```

```
no maximum-paths
```

Description

Sets the maximum number of ECMP routes that RIP can support.

The `no` form of this command sets the maximum number of ECMP routes to the default value of 4.

Command context

config

Parameters

<MAX-VALUE>

Sets the number of RIP ECMP routes. Range: 1-8.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting maximum number of RIP ECMP routes:

```
switch(config)# router rip 1
switch (config-rip-1)# maximum-paths 8
```

Setting maximum number of RIP ECMP routes to default:

```
switch(config)# router rip 1
switch (config-rip-1)# no maximum-paths
```

redistribute

Syntax

```
redistribute {bgp | connected | ospf <PROCESS-ID> | static}
```

```
no redistribute {bgp | connected | ospf <PROCESS-ID> | static}
```

Description

Redistributes routes originating from other protocols into RIP.

The `no` form of this command disables redistribution of routes originating from other protocols into RIP.

Command context

config

Parameters

bgp

Specifies BGP routes to redistribute into RIP.

connected

Specifies connected routes (directly attached subnet or host) to redistribute into RIP.

ospf <PROCESS-ID>

Specifies OSPF route to redistribute into RIP.

static

Specifies static route to redistribute into RIP.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Redistributing BGP routes into RIP:

```
switch(config)# router rip 1  
switch(config-rip-1)# redistribute bgp
```

Disabling BGP routes that originate from other protocols and redistribute into RIP:

```
switch(config)# router rip 1  
switch(config-rip-1)# no redistribute bgp
```

timers update

Syntax

```
timers update <INTERVAL> timeout <DURATION> garbage-collection <PERIOD>
```

```
no timers
```

Description

Configures RIP timers with specific values.

The `no` form of this command sets all RIP timers to default values.

Command context

```
config
```

Parameters

timers update <INTERVAL>

Specifies frequency at which RIP sends updates to all of its peers. Range: 1 to 2147484. Default: 30.

timeout <DURATION>

Specifies timeout duration from the point of the last refresh after a route is received from a peer timeout and is marked as expired. Range: 1 to 255. Default: 180.

garbage-collection <PERIOD>

Specifies amount of time route remains in routing table after route expiration. Range: 1 to 255. Default: 120.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring RIP timers with specific values:

```
switch(config)# router rip 1  
switch(config-rip-1)# timers update 40 timeout 200 garbage-collection 150
```

Configuring RIP timers with default values:


```
switch(config)# router rip 1  
switch(config-rip-1)# no timers
```

RIPv2 clear commands

clear ip rip statistics

Syntax

```
clear ip rip [<PROCESS-ID>] statistics [all-vrfs | vrf <VRF-NAME>]
```

Description

Clears RIP event statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63

all-vrfs

Clears statistics for all VRFs.

vrf

Selects VRF to clear statistics for.

<VRF-NAME>

Specifies VRF name.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Clearing RIP event statistics:

```
switch# clear ip rip statistics
```

RIPv2 interface commands

enable

Syntax

```
enable
```

```
no enable
```

Description

Enables RIP process for RIP enabled IP address configured on interface.

The **no** form of this command disables RIP process on interface.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Enabling RIP process for RIP enabled IP address:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip 1 10.1.1.1
switch(config-if-rip)# enable
```

Disabling RIP process on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip 1 10.1.1.1
switch(config-if-rip)# no enable
```

disable

Syntax

disable

no disable

Description

Disables RIP process for RIP enabled IP addresses configured on interface.

The `no` form of this command enables RIP process on interface.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling RIP process for RIP enabled IP addresses configured on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip 1 10.1.1.1
switch(config-if-rip)# disable
```

Enabling RIP process on interface:

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip 1 10.1.1.1  
switch(config-if-rip)# no disable
```

send disable

Syntax

```
send disable
```

```
no send disable
```

Description

Disables an interface from sending RIP packets for a specific IP address.

The `no` form of this command enables interface for sending RIP packets for a specific IP address.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to send and receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Disabling interface from sending RIP packets for a specific IP address :

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip 1 10.1.1.1  
switch(config-if-rip)# send disable
```

Enabling interface to send RIP packets for a specific IP address:

```
switch(config)# interface 1/1/1  
switch(config-if)# ip rip 1 10.1.1.1  
switch(config-if-rip)# no send disable
```

receive disable

Syntax

```
receive disable
```

```
no receive disable
```

Description

Disables interface from receiving RIP packets for a specific IP address.

The `no` form of this command enables interface for receiving RIP packets for a specific IP address.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Default settings allow an interface to receive RIP packets.
- If an IP address is removed from an interface configured with RIP, all RIP configurations will be removed from the interface.

Examples

Disabling interface from receiving RIP packets for a specific IP address:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip 1 10.1.1.1
switch(config-if-rip)# receive disable
```

Enabling interface for receiving RIP packets for a specific IP address:

```
switch(config)# interface 1/1/1
switch(config-if)# ip rip 1 10.1.1.1
switch(config-if-rip)# no receive disable
```

RIPv2 show commands

show capacities rip

Syntax

show capacities rip

Description

Displays maximum number of RIP interfaces, routes and process.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying maximum number of RIP interfaces, routes and process:

```
switch# show capacities rip

System Capacities: Filter RIP
Capacities Name                                     Value
-----
Maximum number of RIP interfaces configurable in the system 32
Maximum number of RIP processes supported across each VRF   1
Maximum number of routes in RIP supported across all VRFs   2540
```

show capacities-status rip

Syntax

```
show capacities-status rip
```

Description

Displays number of RIP interfaces, routes and process configured in the system.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying number of RIP interfaces, routes and process:

```
switch# show capacities-status rip

System Capacities Status: Filter RIP
Capacities Name                                     Value Maximum
-----
Number of RIP interfaces configured in the system    0          32
```

show ip rip

Syntax

```
show ip rip [<PROCESS-ID>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays general RIP configuration.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

all vrfs

Displays general RIP information for all VRFs.

vrf

Selects VRF to display general RIP information for.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display general RIP information for a specific RIP process.
- Parameters display general RIP information for a specific or all VRFs.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying general RIP configuration for all VRFs:

```
switch# show ip rip 34 all-vrfs
VRF : Default                               Process-ID : 34
-----
RIP Version           : RIPv2               Protocol Status : Enabled
Update Time          : 60 sec               Timeout Time    : 240 sec
Garbage Collection Time : 250 sec            ECMP           : 6
Distance             : 100                 Redistribution : static,
                                           ospf 1

VRF : vrf_1                               Process-ID : 34
-----
RIP Version           : RIPv2               Protocol Status : Enabled
Update Time          : 30 sec               Timeout Time    : 180 sec
Garbage Collection Time : 120 sec            ECMP           : 4
Distance             : 120                 Redistribution : None
```

show ip rip interface

Syntax

```
show ip rip [<PROCESS-ID>] interface [<INTERFACE-NAME>] [brief] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays information about RIP enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

<INTERFACE-NAME>

Specifies interface.

brief

Shows brief overview information for the RIP interface.

all-vrfs

Displays interface information for all VRFs.

vrf

Selects specific VRF.

<VRF-NAME>

Specifies VRF.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display general RIP information for a specific RIP process.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

```
switch# show ip rip interface
Interface 1/1/1 is up, IP Address is 10.10.10.1/24
-----
VRF           : Default           Process-ID    : 1
Status        : Oper Up           Mode          : Send and Receive
MTU           : 500               Version       : RIPv2
Poision Reverse : Enabled

Interface 1/1/2 is up, IP Address is 20.10.10.1/24
-----
VRF           : Default           Process-ID    : 1
Status        : Admin Down        Mode          : Receive
MTU           : 500               Version       : RIPv2
Poision Reverse : Enabled

Interface 1/1/3 is up, IP Address is 30.10.10.1/24
-----
VRF           : Default           Process-ID    : 1
Status        : Admin Down        Mode          : Send
MTU           : 500               Version       : RIPv2
Poision Reverse : Enabled

switch# show ip rip interface brief
VRF : default   Process-ID : 1
-----

Total Number of Interfaces: 2

Interface      IP-Address/Mask   Status   MTU
-----
1/1/1         10.10.10.1/24    up       500
1/1/2         20.10.10.1/24    up       500
1/1/3         30.10.10.1/24    up       500
```

show ip rip neighbors

Syntax

```
show ip rip [<PROCESS-ID>] neighbors [<IP-ADDRESS>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays information about RIP neighbors.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

<IP-ADDRESS>

Specifies IP address of a specific neighbor to display information on.

all-vrfs

Displays neighbor information for all VRFs.

vrf

Selects VRF to display neighbor information.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display RIP neighbor information for a specific RIP process.
- Parameters display RIP neighbor information for a specific neighbor.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIP neighbor information for all VRFs:

```
switch# show ip rip neighbors all-vrfs
VRF : default          Process-ID : 1
-----

Total Number of Neighbors: 1

Peer-Address    Type      Last-Update  Rcvd-Bad-Pkts  Rcvd-Bad-Routes
-----
1.1.1.2         RIPv2     0:0:7        4               5
```

show ip rip routes

Syntax

```
show ip rip [<PROCESS-ID>] routes [<PREFIX/LENGTH>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIP routing table for a specific RIP process.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID to display information for a specific RIP process. Range: 1-63.

<PREFIX/LENGTH>

Specifies the network prefix.

all-vrfs

Displays RIP routing information for all VRFs.

vrf

Selects VRF to display RIP routing information.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- <PREFIX/LENGTH> is an optional parameter that displays RIP routing table information for a specific subnet.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIP routing table for all VRFs:

```
switch# show ip rip routes all-vrfs
VRF : default                      Process-ID : 1
-----
Total Number of Routes : 6

Prefix          Metric      Interface    Nexthop
-----
10.1.0.0/16      2           1/1/1        30.1.1.2
20.1.2.0/24      3           1/1/1        30.1.1.2
30.1.1.0/24      1           1/1/1

VRF : vrf_1                      Process-ID : 34
-----

Prefix          Metric      Interface    Nexthop
-----
20.1.0.0/16      10          1/1/2        50.1.1.2
40.1.2.0/24      14          1/1/2        50.1.1.2
50.1.1.0/24      1           1/1/2
```

show ip rip statistics

Syntax

```
show ip rip [<PROCESS-ID>] statistics [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIP statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

all-vrfs

Displays statistics information for all VRFs.

vrf

Selects VRF to display RIP statistics information for.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters can display information for all VRFs or a specific VRF.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIP statistics for all VRFs:

```
switch# show ip rip statistics all-vrfs
VRF : default Process-ID : 1
-----

Global Route Changes : 50
Global Queries       : 2
Last Cleared          : 0h 30m 28s ago

VRF : vrf_1 Process-ID : 34
-----

Global Route Changes : 20
Global Queries       : 0
Last Cleared          : 0h 30m 28s ago
```

show ip rip statistics interface

Syntax

```
show ip rip [<PROCESS-ID>] statistics interface [<INTERFACE-NAME>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIP statistics for RIP enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIP process ID. Range: 1-63.

<INTERFACE-NAME>

Specifies name of interface.

all-vrfs

Displays RIP interface statistics for all VRFs.

vrf

Selects VRF to display RIP interface statistics.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters can display information for all VRFs or a specific VRF.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIP statistics for a RIP enabled interface:

```
switch# show ip rip statistics interface 1/1/1
VRF : default Process-ID : 1 interface 1/1/1
-----
IP-Address      Trigger-Updates  Rcvd-Bad-Packets  Rcvd-Bad-Routes
-----
10.1.1.1        15               3                  4
Last Cleared : 0h 30m 28s ago
```

show running-config

Syntax

show running config

Description

Displays all running configurations for all protocols including RIP.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying all running configurations for all protocols including RIP:

```
switch# show running-config
Current configuration:
!
!Version Halon 0.1.0 (Build: genericx86-64-Halon-0.1.0-master-20170309054955-dev)
!Schema version 0.1.8
lldp enable
timezone set utc
vrf blue
vrf green
vrf red
led base-loc_fdc on
led base-loc_on
led base-hlth_fdc fast_blink
led base-pwr_fdc on
!
!
!
!
!
!
aaa authentication login default local
aaa authorization commands default none
!
!
!
!
router ospf 1 vrf red
router rip 1
    maximum-paths 5
    distance 1
router rip 1 vrf red
    default-information originate always
    maximum-paths 7
    distance 5
    redistribute ospf 1
    timers update 40 timeout 200 garbage-collection 120
vlan 1
    no shutdown
interface lag 44
    no shutdown
    ip address 33.1.1.1/24
    ip rip 1 33.1.1.1
        send disable
interface 1/1/1
    no shutdown
    ip address 33.44.1.1/24
    ip address 44.44.1.1/24 secondary
    ip rip 1 33.44.1.1
        send disable
    ip rip 1 44.44.1.1
        send disable
interface 1/1/2
interface loopback 2
```

```
ip address 55.55.55.55/32
ip rip 1 55.55.55.55
```

RIPng (IPv6) commands

Configuration commands

router ripng

Syntax

```
router ripng <PROCESS-ID> [vrf <VRF-NAME>]
```

```
no router ripng <PROCESS-ID> [vrf <VRF-NAME>]
```

Description

Creates RIPng process if not already created and enters the `router ripng <PROCESS-ID>` context for the VRF mentioned. If no VRF is mentioned, a default is used. Only one RIPng process is allowed per VRF.

The `no` form of this command deletes the RIPng instance for the VRF. If no VRF is mentioned the default is deleted.

Command context

config

Parameters

<PROCESS-ID>

Specifies name of the RIPng process ID. Range: 1-63.

vrf

Sets VRF name for RIPng process.

<VRF-NAME>

VRF name for VRF.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating RIPng process and naming the VRF:

```
switch(config)# router ripng 2 vrf red
```

Deleting RIPng process:

```
switch(config)# no router ripng 2 vrf red
```

Interface commands

ipv6 ripng

Syntax

```
ipv6 ripng <PROCESS-ID>
```

```
no ipv6 ripng <PROCESS-ID>
```

Description

Enables RIPng process on interface and creates a new context.

The `no` form of this command deletes RIPng process on interface.

Command context

```
config-if
```

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling RIPng process on an interface:

```
switch(config)# interface 1/1/1  
switch (config-if)# ipv6 ripng 1  
switch (config-if-ripng)#
```

Deleting RIPng process on an interface:

```
switch(config)# interface 1/1/1  
switch (config-if)# no ipv6 ripng 1
```

Routing commands

enable

Syntax

```
enable
```

```
no enable
```

Description

Enables RIPng process if disabled. By default RIPng process is enabled.

The `no` form of this command disables the RIPng process.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling RIPng process when disabled:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# enable
```

Disabling RIPng process when enabled:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# no enable
```

disable

Syntax

disable

no disable

Description

Disables RIPng process.

The `no` form of this command enables the RIPng process.

Command context

config

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling RIPng process:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# disable
```

Enabling RIPng process:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# no disable
```

distance

Syntax

distance <DISTANCE>

no distance

Description

Configures administrative distance for RIPng. Administrative distance is used as criteria to select the best route when multiple protocols have the same route.

The `no` form of this command sets the RIPng administrative distance to the default. Default: 120.

Command context

config

Parameters

<DISTANCE>

Specifies RIPng administrative distance. Range: 1 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring administrative distance for RIPng:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# distance 100
```

Setting administrative distance for RIPng to default values:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# no distance
```

maximum-paths

Syntax

maximum-paths **<MAX-VALUE>**

no maximum-paths

Description

Sets the maximum number of ECMP routes that RIPng can support.

The `no` form of this command sets the maximum number of ECMP routes to the default value of 4.

Command context

config

Parameters

<MAX-VALUE>

Sets the number of RIPng ECMP routes. Range: 1-8.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting maximum number of RIPng ECMP routes:

```
switch(config)# router ripng 1  
switch (config-ripng-1)# maximum-paths 8
```

Setting maximum number of RIPng ECMP routes to default:


```
switch(config)# router ripng 1  
switch (config-ripng-1)# no maximum-paths
```

redistribute

Syntax

```
redistribute {bgp | connected | ospfv3 <PROCESS-ID> | static}  
  
no redistribute {bgp | connected | ospfv3 <PROCESS-ID> | static}
```

Description

Redistributes routes originating from other protocols into RIPng.

The **no** form of this command disables redistribution of routes originating from other protocols into RIPng.

Command context

config

Parameters

bgp

Specifies BGP routes to redistribute into RIPng.

connected

Specifies connected routes (directly attached subnet or host) to redistribute into RIPng.

ospfv3 <PROCESS-ID>

Specifies OSPFv3 route to redistribute into RIPng.

static

Specifies static route to redistribute into RIPng.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Redistributing BGP routes into RIPng:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# redistribute bgp
```

Disabling BGP routes that originate from other protocols and redistribute into RIPng:

```
switch(config)# router ripng 1  
switch(config-ripng-1)# no redistribute bgp
```

timers update

Syntax

```
timers update <INTERVAL> timeout <DURATION> garbage-collection <PERIOD>  
  
no timers
```

Description

Configures RIPng timers with specific values.

The `no` form of this command sets all RIPng timers to default values.

Command context

`config`

Parameters

`timers update <INTERVAL>`

Specifies frequency at which RIPng sends updates to all of its peers. Range: 1 to 2147484. Default: 30.

`timeout <DURATION>`

Specifies timeout duration from the point of the last refresh after a route is received from a peer timeout and is marked as expired. Range: 1 to 255. Default: 180.

`garbage-collection <PERIOD>`

Specifies amount of time route remains in routing table after route expiration. Range: 1 to 255. Default: 120.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring RIPng timers with specific values:

```
switch(config)# router ripng 1
switch(config-ripng-1)# timers update 40 timeout 200 garbage-collection 150
```

Configuring RIPng timers with default values:

```
switch(config)# router ripng 1
switch(config-ripng-1)# no timers
```

RIPng clear commands

`clear ipv6 ripng statistics`

Syntax

```
clear ipv6 ripng [<PROCESS-ID>] statistics [all-vrfs | vrf <VRF-NAME>]
```

Description

Clears RIPng event statistics.

Command context

Operator (>) or Manager (#)

Parameters

`<PROCESS-ID>`

Specifies RIPng process ID. Range: 1-63

all-vrfs

Clears statistics for all VRFs.

vrf

Selects VRF to clear statistics for.

<VRF-NAME>

Specifies VRF name.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Clearing RIPng event statistics:

```
switch# clear ipv6 ripng statistics
```

RIPng interface commands

enable

Syntax

enable

no enable

Description

Enables RIPng process on interface.

The **no** form of this command disables RIPng process on interface.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling RIPng process on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ripng 1
switch(config-if-ripng)# enable
```

Disabling RIPng process on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ripng 1
switch(config-if-ripng)# no enable
```

disable

Syntax

disable

no disable

Description

Disables RIPng process on interface.

The `no` form of this command enables RIPng process on interface.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling RIP process for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ripng 1
switch(config-if-ripng)# disable
```

Enabling RIP process for all RIP enabled IP addresses on interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 ripng 1
switch(config-if-ripng)# no disable
```

send disable

Syntax

send disable

no send disable

Description

Disables interface from sending RIPng packets. An interface can send RIPng packets by default.

The `no` form of this command enables interface to send RIPng packets, if disabled.

Command context

config-if

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling interface from sending RIPng packets:

```
switch(config)# interface 1/1/1  
switch (config-if)# ipv6 ripng 1  
switch (config-if-ripng)# send disable
```

Enabling interface to send RIPng packets:

```
switch(config)# interface 1/1/1  
switch (config-if)# ipv6 ripng 1  
switch (config-if-ripng)# no send disable
```

receive disable

Syntax

```
receive disable
```

```
no receive disable
```

Description

Disables interface from receiving RIPng packets for all enabled IP addresses. An interface can receive RIPng packets by default.

The `no` form of this command enables interface to receive RIPng packets, if disabled.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Disabling interface from receiving RIPng packets:

```
switch(config)# interface 1/1/1  
switch(config-if)# ipv6 ripng 1  
switch (config-if-ripng)# receive disable
```

Enabling interface to receive RIPng packets when disabled:

```
switch(config)# interface 1/1/1  
switch(config-if)# ipv6 ripng 1  
switch (config-if-ripng)# no receive disable
```

RIPng show commands

show capacities ripng

Syntax

```
show capacities ripng
```

Description

Displays the maximum number of RIPng interfaces, routes and process.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying maximum number of RIPng interfaces, routes and process:

```
switch# show capacities ripng

System Capacities: Filter RIPng
Capacities Name                                     Value
-----
Maximum number of RIPng interfaces configurable in the system 32
Maximum number of RIPng processes supported across each VRF    1
Maximum number of routes in RIPng supported across all VRFs    2540
```

show capacities-status ripng

Syntax

```
show capacities-status ripng
```

Description

Displays number of RIPng interfaces, routes and process configured in the system.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying number of RIPng interfaces, routes and process:

```
switch# show capacities-status ripng

System Capacities Status: Filter RIPng
Capacities Name                                     Value  Maximum
-----
Number of RIPng interfaces configured in the system      0       32
```

show ipv6 ripng

Syntax

```
show ipv6 ripng [<PROCESS-ID>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays general RIPng configuration.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

all vrfs

Displays general RIPng information for all VRFs.

vrf

Selects VRF to display general RIPng information for.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display general RIPng information for a specific RIPng process.
- Parameters display general RIPng information for a specific or all VRFs.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying general RIPng configuration for all VRFs:

```
switch# show ipv6 ripng 34 all-vrfs
VRF : Default                               Process-ID : 34
-----
Protocol Status      : Enabled   ECMP           : 6
Update Time         : 60 sec    Timeout Time    : 240 sec
Garbage Collection Time : 250 sec Distance       : 100
Redistribution       : static,
                    ospfv3 1

VRF : vrf_1                               Process-ID : 34
-----
Protocol Status      : Enabled   ECMP           : 4
Update Time         : 30 sec    Timeout Time    : 180 sec
Garbage Collection Time : 120 sec Distance       : 120
Redistribution       : None
```

show ipv6 ripng interface

Syntax

```
show ipv6 ripng [<PROCESS-ID>] interface [<INTERFACE-NAME>] [brief] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays information about RIPng enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

<INTERFACE-NAME>

Specifies interface.

brief

Shows brief overview information for RIPng interface.

all-vrfs

Displays interface information for all VRFs.

vrf

Selects specific VRF.

<VRF-NAME>

Specifies VRF.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display general RIPng information for a specific RIPng process.
- Parameters display general RIPng information for a specific or all VRFs.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

```
switch# show ipv6 ripng interface
Interface 1/1/1 is up, IPv6 Address is fe80::7272:cfff:fe70:67a
-----
VRF           : Default           Process-ID      : 1
Status        : Oper Up           Mode            : Send and Receive
MTU           : 500               Poision Reverse : Enabled

Interface 1/1/2 is up, IPv6 Address is fe80::7272:cfff:fe70:67a
-----
VRF           : Default           Process-ID      : 1
Status        : Admin Down        Mode            : Receive
MTU           : 500               Poision Reverse : Enabled

Interface 1/1/3 is up, IPv6 Address is fe80::7272:cfff:fe70:67a
-----
VRF           : Default           Process-ID      : 1
Status        : Admin Down        Mode            : Send
MTU           : 500               Poision Reverse : Enabled

switch# show ipv6 ripng interface brief
VRF : default   Process-ID : 1
-----

Total Number of Interfaces: 2

Interface   IPv6-Address           Status   MTU
-----
1/1/1      fe80::7272:cfff:fe70:67a   up       500
1/1/2      fe80::7272:cfff:fe71:67a   up       500
```


show ipv6 ripng neighbors

Syntax

```
show ipv6 ripng [<PROCESS-ID>] neighbors [<LINK-LOCAL-ADDRESS>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays information about RIPng neighbors.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

neighbors

Specifies neighbor IP address.

<LINK-LOCAL-ADDRESS>

Specifies link-local address.

all-vrfs

Displays neighbor information for all VRFs.

vrf

Selects VRF to display neighbor information.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters display RIPng neighbor information for a specific RIPng process.
- Parameters display RIPng neighbor information for a specific neighbor.
- Parameters display general RIPng information for a specific or all VRFs.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIPng neighbor information for all VRFs:

```
switch# show ipv6 ripng neighbors all-vrfs
VRF : default          Process-ID : 1
-----

Total Number of Neighbors: 1

Peer-Address      Type      Last-Update  Rcvd-Bad-Pkts  Rcvd-Bad-Routes
-----
```

show ipv6 ripng routes

Syntax

```
show ipv6 ripng [<PROCESS-ID>] routes [<PREFIX/LENGTH>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIPng routing table for a specific RIPng process.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID to display information for a specific RIPng process. Range: 1-63.

<PREFIX/LENGTH>

Specifies the network prefix.

all-vrfs

Displays RIPng routing information for all VRFs.

vrf

Selects VRF to display RIPng routing information.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- <PREFIX/LENGTH> is an optional parameter that displays RIPng routing table information for a specific subnet.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIPng routing table for all VRFs:

```
switch# show ipv6 ripng routes all-vrfs
```

```
VRF : default Process-ID : 1
```

Prefix	Metric	Interface	Nexthop
2001:DB8:10::/64	2	1/1/1	FE80::2E0:E6FF:FE1B:8242
2002:DB8:10::/64	3	1/1/1	FE80::2E0:E6FF:FE1B:8242
2003:DB8:10::/64	1	1/1/1	

```
VRF : vrf_1 Process-ID : 34
```

Prefix	Metric	Interface	Nexthop
3001:DB8:10::/64	10	1/1/2	FE80::2E0:E6FF:FE1B:8232
3002:DB8:10::/64	14	1/1/2	FE80::2E0:E6FF:FE1B:8232
3003:DB8:10::/64	1	1/1/2	

show ipv6 ripng statistics

Syntax

```
show ipv6 ripng [<PROCESS-ID>] statistics [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIPng statistics.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

all-vrfs

Displays statistics information for all VRFs.

vrf

Selects VRF and displays RIPng statistics for it.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters can display information for all VRFs or a specific VRF.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIPng statistics for all VRFs:

```
switch# show ipv6 ripng statistics all-vrfs
VRF : default Process-ID : 1
-----
Global Route Changes : 50
Global Queries       : 2
Last Cleared          : 0h 30m 28s ago

VRF : vrf_1 Process-ID : 34
-----
```

```
Global Route Changes : 20
Global Queries       : 0
Last Cleared          : 0h 30m 28s ago
```

show ipv6 ripng statistics interface

Syntax

```
show ipv6 ripng [<PROCESS-ID>] statistics interface [<INTERFACE-NAME>] [all-vrfs | vrf <VRF-NAME>]
```

Description

Displays RIPng statistics for RIPng enabled interfaces.

Command context

Operator (>) or Manager (#)

Parameters

<PROCESS-ID>

Specifies RIPng process ID. Range: 1-63.

<INTERFACE-NAME>

Specifies name of interface.

all-vrfs

Displays RIPng interface statistics for all VRFs.

vrf

Selects VRF to display RIPng interface statistics.

<VRF-NAME>

Specifies VRF name.

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Usage

- Parameters can display information for all VRFs or a specific VRF.
- If a VRF is not mentioned, information for the default VRF is displayed.

Examples

Displaying RIPng statistics for a RIPng enabled interface:

```
switch# show ipv6 ripng statistics interface 1/1/1
VRF : default Process-ID : 1 interface 1/1/1
```

IPv6-Address	Trigger-Updates	Rcvd-Bad-Packets	Rcvd-Bad-Routes
fe80::7272:cfff:fe70:86ae	15	3	4

show running-config

Syntax

```
show running config
```

Description

Displays all running configurations for all protocols including RIPng.

Command context

Operator (>) or Manager (#)

Authority

Auditors or Administrators or local user group members with execution rights for this command. Auditors can execute this command from the auditor context (auditor>) only.

Examples

Displaying all running configurations for all protocols including RIPng:

```
switch# show running-config
Current configuration:
!
!Version Halon 0.1.0 (Build: genericx86-64-Halon-0.1.0-master-20170309054955-dev)
!Schema version 0.1.8
lldp enable
timezone set utc
vrf green
vrf red
led base-loc_fdc on
led base-loc on
led base-hlth_fdc fast_blink
led base-pwr_fdc on
!
!
!
!
!
!
aaa authentication login default local
aaa authorization commands default none
!
!
!
!
router ospfv3 1
router ripng 1
    maximum-paths 5
    distance 1
    redistribute ospfv3 1
    timers update 40 timeout 200 garbage-collection 150
router ripng 1 vrf red
    default-information originate always
    maximum-paths 7
    distance 5
vlan 1
    no shutdown
```

```
interface lag 44
  no shutdown
  ipv6 address link-local
  ipv6 ripng 1
    send disable
interface 1
  no shutdown
  ipv6 address link-local
  ipv6 ripng 1
    receive disable
```

Accessing Aruba Support

Aruba Support Services	https://www.arubanetworks.com/support-services/
Aruba Support Portal	https://asp.arubanetworks.com/
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	https://www.arubanetworks.com/support-services/contact-support/

Be sure to collect the following information before contacting Support:

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Other useful sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	https://community.arubanetworks.com/
Software licensing	https://lms.arubanetworks.com/
End-of-Life information	https://www.arubanetworks.com/support-services/end-of-life/
Aruba software and documentation	https://asp.arubanetworks.com/downloads

Accessing updates

To download product updates:

Aruba Support Portal

<https://asp.arubanetworks.com/downloads>

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

<https://www.hpe.com/networking/support>

To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

<https://support.hpe.com/portal/site/hpsc/aae/home/>



IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

To subscribe to eNewsletters and alerts:

<https://asp.arubanetworks.com/notifications/subscriptions> (requires an active Aruba Support Portal (ASP) account to manage subscriptions). Security notices are viewable without an ASP account.

Warranty information

To view warranty information for your product, go to <https://www.arubanetworks.com/support-services/product-warranties/>.

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at <https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see <https://www.arubanetworks.com/company/about-us/environmental-citizenship/>.

Documentation feedback

Aruba is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback-switching@hpe.com). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.