

BGP & BGP4+ (Border Gateway Protocol IPv4 & IPv6) Software Reference Supplement for x-Series Switches **AlliedWare Plus™ Operating System** **Version 5.4.3-2.6**



SwitchBlade® x8112

SwitchBlade® x908

x900-24XS and x900-24XT

x900-12XT/S

x610-24Ts and x610-24Ts/X

x610-48Ts and x610-48Ts/X

x610-24Ts-PoE+ and x610-24Ts/X-PoE+

x610-48Ts-PoE+ and x610-48Ts/X-PoE+

x610-24SPs/X

Acknowledgments

This product includes software developed by the University of California, Berkeley and its contributors.

Copyright ©1982, 1986, 1990, 1991, 1993 The Regents of the University of California.
All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

Copyright ©1998-2008 The OpenSSL Project. All rights reserved.

This product includes software licensed under the GNU General Public License available from:

<http://www.gnu.org/licenses/gpl2.html>

Source code for all GPL licensed software in this product can be obtained from the Allied Telesis GPL Code Download Center at:

<http://www.alliedtelesis.com/support/default.aspx>

Allied Telesis is committed to meeting the requirements of the open source licenses including the GNU General Public License (GPL) and will make all required source code available.

If you would like a copy of the GPL source code contained in Allied Telesis products, please send us a request by registered mail including a check for US\$15 to cover production and shipping costs and a CD with the GPL code will be mailed to you.

GPL Code Request
Allied Telesis Labs (Ltd)
PO Box 8011
Christchurch.
New Zealand

©2013 Allied Telesis Inc. All rights reserved.

This documentation is subject to change without notice. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's internal use without the written permission of Allied Telesis, Inc.

Allied Telesis, AlliedWare Plus, EPSRing, SwitchBlade, and VCStack are trademarks or registered trademarks in the United States and elsewhere of Allied Telesis, Inc. Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Additional brands, names and products mentioned herein may be trademarks of their respective companies.

Getting the most from this manual

Although you can view this document using Acrobat version 5, to get the best from this manual, we recommend using Adobe Acrobat Reader version 8. You can download Acrobat Reader 8 free from <http://www.adobe.com/>.

New features in this software version

For a list of new and enhanced features and commands in this version, see the Software Release Note for Software Version 5.4.3-2.6. Documentation can be downloaded from the Support area of our website at <http://www.alliedtelesis.com>. Note that to download software files, you need a valid user account.

Table of Contents



Chapter 1 BGP and BGP4+ Introduction

Introduction.....	1.3
BGP and BGP4+ ASNs and AS_Path Attributes.....	1.4
Internal and External BGP Concepts.....	1.5
Public and Private ASNs	1.5
Outbound Routing Toward the Internet.....	1.5
External BGP (eBGP) Concepts.....	1.6
External BGP (eBGP).....	1.6
Verifying BGP.....	1.7
Advertising eBGP routes to ISPs.....	1.8
Internal BGP (iBGP) Concepts.....	1.9
iBGP between Internet connected routers.....	1.9
Configuring iBGP	1.9
Verifying iBGP	1.10
How to configure BGP and BGP4+	1.12
Initial BGP and BGP4+ Configuration Tasks	1.12
How to enable BGP and BGP4+ Routing.....	1.12
How to configure BGP and BGP4+ Neighbors	1.12
BGP and BGP4+ Neighbor Route Filtering	1.13
BGP and BGP4+ Path Filtering by Neighbor	1.13
How to disable next hop processing for BGP or BGP4+ updates.....	1.14
How to use Route Maps for BGP or BGP4+ updates	1.15
BGP and BGP4+ Peer Groups	1.16
How to create a Peer Group for BGP.....	1.16
How to assign options to the created Peer Group	1.16
How to make neighbors members of a Peer Group	1.19
BGP and BGP4+ Prefix Lists.....	1.20
How to create and configure a Prefix List for BGP or BGP4+	1.21
How to delete Prefix List Entries.....	1.21
How to display IPv4 and IPv6 Prefix Entries	1.22
BGP and BGP4+ Route Dampening	1.23
How to enable BGP and BGP4+ Route Dampening.....	1.24
How to monitor BGP and BGP4+ Route Dampening	1.24
BGP and BGP4+ Synchronization	1.26
BGP and BGP4+ Weights.....	1.27
BGP and BGP4+ Aggregate Addresses	1.28
How to enable and disable Automatic Summarization.....	1.28
BGP and BGP4+ monitoring	1.29
How to clear BGP and BGP4+ tables.....	1.29
How to display BGP and BGP4+ statistics	1.30
BGP Neighbor status logging.....	1.31
BGP Connections Reset	1.32
BGP Versions.....	1.32
BGP Confederations.....	1.32
BGP Route Reflectors.....	1.33
BGP Community Filtering	1.34

BGP Optimization	1.35
How to determine Backdoor Routes	1.35
How to set the BGP Administrative Distance	1.35
How to change BGP default local preference value	1.35
How to redistribute the default network 0.0.0.0	1.36
How to adjust BGP Timers	1.36
How to adjust the eBGP advertisement interval	1.37
How to adjust the iBGP advertisement interval	1.37
How to adjust the AS origination interval	1.38
How to remove private AS numbers for BGP	1.39
BGP and BGP4+ Terminology	1.42

Chapter 2 BGP and BGP4+ Configuration

Introduction.....	2.2
BGP Configurations.....	2.3
Enabling BGP Peers In The Same Autonomous System	2.3
Enabling BGP Between Different Autonomous Systems	2.5
Configuring a BGP Route Reflector	2.6
Configuring a BGP Confederation	2.8
Configuring BGP Authentication	2.11
Configuring BGP Graceful Reset.....	2.13
Configuring BGP Distance	2.15
Configuring BGP Graceful Restart.....	2.18
Configuring BGP Weight per Peer	2.20
BGP4+ Configurations	2.23
Enabling iBGP Peering using a Global Address	2.23
Enabling iBGP Peering using a Link-local Address	2.26
Enabling eBGP Peering between different Autonomous Systems.....	2.29
Configuring Route-Maps with BGP4+.....	2.31
Configuring BGP4+ Graceful Restart	2.35
Configuring BGP4+ Distance.....	2.39
Configuring BGP4+ Graceful Reset	2.43

Chapter 3 BGP and BGP4+ Commands

Introduction	3.5
Controlling "show" Command Output.....	3.6
Command List	3.8
address-family (BGP and BGP4+)	3.8
aggregate-address (BGP and BGP4+)	3.10
auto-summary (BGP).....	3.12
bgp aggregate-nexthop-check (BGP and BGP4+)	3.13
bgp always-compare-med (BGP and BGP4+)	3.14
bgp bestpath as-path ignore (BGP and BGP4+)	3.15
bgp bestpath compare-confed-aspath (BGP and BGP4+)	3.15
bgp bestpath compare-routerid (BGP and BGP4+)	3.16
bgp bestpath med (BGP and BGP4+)	3.17
bgp bestpath med remove-recv-med (BGP and BGP4+)	3.18
bgp bestpath med remove-send-med (BGP and BGP4+)	3.18
bgp client-to-client reflection (BGP and BGP4+)	3.19
bgp cluster-id (BGP and BGP4+).....	3.20
bgp confederation identifier (BGP and BGP4+)	3.21
bgp confederation peers (BGP and BGP4+)	3.22
bgp config-type (BGP and BGP4+)	3.24
bgp dampening (BGP and BGP4+)	3.26
bgp default ipv4-unicast (BGP and BGP4+).....	3.27
bgp default local-preference (BGP)	3.28

bgp deterministic-med (BGP and BGP4+)	3.29
bgp enforce-first-as (BGP and BGP4+).....	3.30
bgp fast-external-failover (BGP and BGP4+)	3.31
bgp graceful-restart (BGP and BGP4+).....	3.32
bgp graceful-restart graceful-reset (BGP and BGP4+)	3.33
bgp log-neighbor-changes (BGP and BGP4+).....	3.34
bgp memory maxallocation (BGP and BGP4+)	3.36
bgp nexthop-trigger-count (BGP and BGP4+)	3.37
bgp nexthop-trigger delay (BGP and BGP4+).....	3.38
bgp nexthop-trigger enable (BGP and BGP4+).....	3.39
bgp rfc1771-path-select (BGP)	3.40
bgp rfc1771-strict (BGP).....	3.40
bgp router-id (BGP and BGP4+).....	3.41
bgp scan-time (BGP)	3.42
bgp update-delay (BGP and BGP4+).....	3.43
clear bgp * (BGP and BGP4+)	3.44
clear bgp (IPv4 or IPv6 address) (BGP and BGP4+)	3.45
clear bgp (ASN) (BGP and BGP4+).....	3.47
clear bgp external (BGP and BGP4+).....	3.48
clear bgp peer-group (BGP and BGP4+)	3.49
clear ip bgp * (BGP).....	3.50
clear ip bgp (IPv4) (BGP)	3.52
clear ip bgp dampening (BGP)	3.53
clear ip bgp flap-statistics (BGP)	3.53
clear ip bgp (ASN) (BGP)	3.54
clear ip bgp external (BGP)	3.55
clear ip bgp peer-group (BGP).....	3.56
clear bgp ipv6 (ipv6 address) (BGP4+)	3.57
clear bgp ipv6 dampening (BGP4+).....	3.58
clear bgp ipv6 flap-statistics (BGP4+)	3.58
clear bgp ipv6 (ASN) (BGP4+)	3.59
clear bgp ipv6 external (BGP4+)	3.60
clear bgp ipv6 peer-group (BGP4+)	3.61
debug bgp (BGP).....	3.62
distance (BGP and BGP4+).....	3.63
exit-address-family (BGP and BGP4+).....	3.65
ip as-path access-list (BGP and BGP4+).....	3.66
ip community-list (BGP and BGP4+).....	3.67
ip community-list expanded (BGP and BGP4+)	3.68
ip community-list standard (BGP and BGP4+).....	3.70
ip extcommunity-list expanded (BGP and BGP4+)	3.72
ip extcommunity-list standard (BGP and BGP4+)	3.74
ip prefix-list (IPv4 Prefix List)	3.76
ipv6 prefix-list (IPv6 Prefix List)	3.78
match as-path (Route Map).....	3.80
match community (Route Map).....	3.81
neighbor activate (BGP and BGP4+)	3.82
neighbor advertisement-interval (BGP and BGP4+)	3.85
neighbor allowas-in (BGP and BGP4+).....	3.88
neighbor as-origination-interval (BGP and BGP4+).....	3.91
neighbor attribute-unchanged (BGP and BGP4+)	3.93
neighbor capability graceful-restart (BGP and BGP4+).....	3.96
neighbor capability orf prefix-list (BGP and BGP4+)	3.99
neighbor capability route-refresh (BGP and BGP4+)	3.102
neighbor collide-established (BGP and BGP4+)	3.105
neighbor default-originate (BGP and BGP4+)	3.107
neighbor description (BGP and BGP4+).....	3.110

neighbor disallow-infinite-holdtime (BGP and BGP4+)	3.112
neighbor distribute-list (BGP and BGP4+).....	3.114
neighbor dont-capability-negotiate (BGP and BGP4+).....	3.117
neighbor ebgp-multihop (BGP and BGP4+).....	3.119
neighbor enforce-multihop (BGP and BGP4+).....	3.122
neighbor filter-list (BGP and BGP4+).....	3.125
neighbor interface (BGP4+).....	3.128
neighbor local-as (BGP and BGP4+)	3.129
neighbor maximum-prefix (BGP and BGP4+)	3.131
neighbor next-hop-self (BGP and BGP4+)	3.134
neighbor override-capability (BGP and BGP4+)	3.137
neighbor passive (BGP and BGP4+)	3.139
neighbor password (BGP and BGP4+).....	3.141
neighbor peer-group (add a neighbor) (BGP and BGP+)	3.144
neighbor peer-group (create a peer-group) (BGP and BGP4+).....	3.146
neighbor port (BGP and BGP4+).....	3.147
neighbor prefix-list (BGP and BGP4+)	3.149
neighbor remote-as (BGP and BGP4+).....	3.152
neighbor remove-private-AS (BGP).....	3.155
neighbor restart-time (BGP and BGP4+).....	3.156
neighbor route-map (BGP and BGP4+).....	3.159
neighbor route-reflector-client (BGP).....	3.164
neighbor route-server-client (BGP)	3.165
neighbor send-community (BGP and BGP4+).....	3.166
neighbor shutdown (BGP and BGP4+).....	3.170
neighbor soft-reconfiguration inbound (BGP and BGP4+).....	3.172
neighbor timers (BGP and BGP4+).....	3.175
neighbor transparent-as (BGP and BGP4+)	3.178
neighbor transparent-nexthop (BGP and BGP4+)	3.180
neighbor unsuppress-map (BGP and BGP4+)	3.182
neighbor update-source (BGP and BGP4+)	3.185
neighbor version (BGP)	3.188
neighbor weight (BGP and BGP4+)	3.190
network (BGP and BGP4+)	3.193
network synchronization (BGP and BGP4+)	3.196
redistribute (into BGP or BGP4+) (BGP and BGP4+)	3.197
restart bgp graceful (BGP).....	3.199
router bgp (BGP and BGP4+)	3.200
route-map (Route Map)	3.201
set as-path (Route Map).....	3.203
set community (Route Map).....	3.204
show bgp ipv6 (BGP4+)	3.205
show bgp ipv6 community (BGP4+)	3.206
show bgp ipv6 community-list (BGP4+).....	3.207
show bgp ipv6 dampening (BGP4+).....	3.208
show bgp ipv6 filter-list (BGP4+).....	3.209
show bgp ipv6 inconsistent-as (BGP4+)	3.209
show bgp ipv6 longer-prefixes (BGP4+).....	3.210
show bgp ipv6 neighbors (BGP4+).....	3.211
show bgp ipv6 paths (BGP4+)	3.214
show bgp ipv6 prefix-list (BGP4+)	3.214
show bgp ipv6 quote-regexp (BGP4+).....	3.215
show bgp ipv6 regexp (BGP4+)	3.216
show bgp ipv6 route-map (BGP4+).....	3.217
show bgp ipv6 summary (BGP4+)	3.217
show bgp nexthop-tracking (BGP)	3.218
show bgp nexthop-tree-details (BGP)	3.218

show bgp memory maxallocation (BGP)	3.218
show debugging bgp (BGP)	3.219
show ip bgp (BGP)	3.220
show ip bgp attribute-info (BGP)	3.221
show ip bgp cidr-only (BGP)	3.222
show ip bgp community (BGP)	3.223
show ip bgp community-info (BGP)	3.224
show ip bgp community-list (BGP)	3.225
show ip bgp dampening (BGP)	3.226
show ip bgp filter-list (BGP)	3.228
show ip bgp inconsistent-as (BGP)	3.229
show ip bgp longer-prefixes (BGP)	3.230
show ip bgp neighbors (BGP)	3.231
show ip bgp neighbors connection-retrytime (BGP)	3.234
show ip bgp neighbors hold-time (BGP)	3.234
show ip bgp neighbors keepalive (BGP)	3.235
show ip bgp neighbors keepalive-interval (BGP)	3.235
show ip bgp neighbors notification (BGP)	3.236
show ip bgp neighbors open (BGP)	3.236
show ip bgp neighbors rcvd-msgs (BGP)	3.237
show ip bgp neighbors sent-msgs (BGP)	3.237
show ip bgp neighbors update (BGP)	3.238
show ip bgp paths (BGP)	3.238
show ip bgp prefix-list (BGP)	3.239
show ip bgp quote-regexp (BGP)	3.240
show ip bgp regexp (BGP)	3.241
show ip bgp route-map (BGP)	3.242
show ip bgp scan (BGP)	3.243
show ip bgp summary (BGP)	3.244
show ip community-list (BGP and BGP4+)	3.245
show ip extcommunity-list (BGP and BGP4+)	3.245
show ip prefix-list (IPv4 Prefix List)	3.246
show ipv6 prefix-list (IPv6 Prefix List)	3.246
show ip protocols bgp (BGP)	3.247
show route-map (Route Map)	3.248
synchronization (BGP and BGP4+)	3.249
timers (BGP and BGP4+)	3.250
undebg bgp (BGP)	3.251

Chapter 1: BGP and BGP4+ Introduction



Introduction	1.3
BGP and BGP4+ ASNs and AS_Path Attributes	1.4
Internal and External BGP Concepts	1.5
Public and Private ASNs.....	1.5
Outbound Routing Toward the Internet.....	1.5
External BGP (eBGP) Concepts	1.6
External BGP (eBGP)	1.6
Verifying BGP	1.7
Advertising eBGP routes to ISPs	1.8
Internal BGP (iBGP) Concepts	1.9
iBGP between Internet connected routers.....	1.9
Configuring iBGP.....	1.9
Verifying iBGP.....	1.10
How to configure BGP and BGP4+.....	1.12
Initial BGP and BGP4+ Configuration Tasks	1.12
How to enable BGP and BGP4+ Routing.....	1.12
How to configure BGP and BGP4+ Neighbors	1.12
BGP and BGP4+ Neighbor Route Filtering.....	1.13
BGP and BGP4+ Path Filtering by Neighbor	1.13
How to disable next hop processing for BGP or BGP4+ updates.....	1.14
How to use Route Maps for BGP or BGP4+ updates	1.15
BGP and BGP4+ Peer Groups.....	1.16
How to create a Peer Group for BGP	1.16
How to assign options to the created Peer Group	1.16
How to make neighbors members of a Peer Group.....	1.19
BGP and BGP4+ Prefix Lists	1.20
How to create and configure a Prefix List for BGP or BGP4+	1.21
How to delete Prefix List Entries.....	1.21
How to display IPv4 and IPv6 Prefix Entries	1.22
BGP and BGP4+ Route Dampening.....	1.23
How to enable BGP and BGP4+ Route Dampening	1.24
How to monitor BGP and BGP4+ Route Dampening.....	1.24
BGP and BGP4+ Synchronization.....	1.26
BGP and BGP4+ Weights	1.27
BGP and BGP4+ Aggregate Addresses.....	1.28
How to enable and disable Automatic Summarization.....	1.28
BGP and BGP4+ monitoring.....	1.29
How to clear BGP and BGP4+ tables.....	1.29
How to display BGP and BGP4+ statistics	1.30
BGP Neighbor status logging	1.31
BGP Connections Reset.....	1.32
BGP Versions	1.32
BGP Confederations	1.32
BGP Route Reflectors	1.33
BGP Community Filtering.....	1.34
BGP Optimization.....	1.35
How to determine Backdoor Routes.....	1.35

How to set the BGP Administrative Distance	1.35
How to change BGP default local preference value.....	1.35
How to redistribute the default network 0.0.0.0	1.36
How to adjust BGP Timers	1.36
How to adjust the eBGP advertisement interval	1.37
How to adjust the iBGP advertisement interval	1.37
How to adjust the AS origination interval	1.38
How to remove private AS numbers for BGP	1.39
BGP and BGP4+ Terminology	1.42

Introduction

This chapter introduces terminology and concepts about BGP for IPv4 and BGP4+ for IPv6, including the concepts of autonomous system numbers (ASN), path attributes (PA), and both internal and external BGP and BGP4+.

For basic BGP and BGP4+ configuration examples, see [Chapter 2, BGP and BGP4+ Configuration](#).

For details about the commands used in these examples, or the outputs from validation commands, see [Chapter 3, BGP and BGP4+ Commands](#). This chapter provides an alphabetical reference of commands used to configure the Border Gateway Protocol for IPv4 (BGP) and for IPv6 (BGP4+).

BGP and BGP4+ Overview

Border Gateway Protocol (BGP for IPv4 and BGP4+ for IPv6) is an exterior gateway protocol (EGP). The purpose of BGP is to advertise, learn, and choose the best paths inside the Internet.

ISPs (Internet Service Providers) use BGP to exchange the Internet routing table with each other. Enterprises also use BGP to exchange routing information with ISPs, allowing the Enterprise routers to learn Internet routes.

RFCs 1771 (BGP4), 1654 (first BGP4 specification), 1105, 1163, 1267 (older version of BGP) describe BGP and BGP4. RFC 2283 Multiprotocol Extensions for BGP-4 describes BGP4+.

BGP not only enables ISPs to exchange routes with each other, but also to control what data passed through their networks. ISPs need to keep fine control over the routes that they advertise out of their network, and who they advertise those networks to. They have commercial reasons for sending different traffic through different paths. In particular, ISPs whose main service is to provide bulk Internet backbone transport need to be very sure whose data they are transporting, as they do not want to be transporting data on behalf of people who have not paid for the service.

Since the paths via which Internet data is directed are subject to commercial agreements, network providers need to be able to implement policies that control the content of their route tables, and control the routes that they advertise to which neighbors. Internal routing protocols like OSPF and RIP do not have facilities for the types of policies that BGP needs. Although some filtering can be performed in OSPF and RIP, the sets of parameters that can be filtered on are rather limited.

So, instead of just using metric as the criterion for choosing the best route to a destination, BGP uses a process with path attributes, where path attributes are a variety of parameters that are associated with routes and exchanged in routing updates. BGP has an elaborate best path algorithm that is controlled by these path attributes, and allows network engineers flexibility in how routers choose the best BGP routes.

Moreover, the routing protocols used between ISPs are advertising huge numbers of routes (potentially hundreds of thousands of routes), so the routing protocol they use needs to be efficient, not a protocol that requires regular updates of all the routes (as the 30 minute refresh OSPF requires). BGP was developed to operate quite differently from OSPF or RIP.

BGP does not send route updates to multiple neighbors in the local subnet (as is typical with IGP), but uses TCP (port 179) to establish connections to just a specific set of peer routers with which it will exchange routing information. BGP peer routers can be in the same subnet, or can be separated by several routers.

BGP does not send any more route updates than it absolutely has to. When routers first peer up, they exchange the route table data that they wish to inform each other of. Thereafter, they only send each other route information if anything changes.

BGP and BGP4+ ASNs and AS_Path Attributes

BGP and BGP4+ uses BGP path attributes (PAs). PAs define information about a path, or route, through a network. Some BGP and BGP4+ PAs describe information that is used to choose the best BGP or BGP4+ route. PAs are also used for other purposes, such as preventing routing loops.

If no BGP or BGP4+ PAs have been explicitly set, BGP and BGP4+ routers use the BGP and BGP4+ AS_Path (autonomous system path) PA when choosing the best route among competing routes. The AS_Path PA itself has many subcomponents, one of which is the autonomous system number (ASN)

The integer ASN identifies one organization that considers itself autonomous from other organizations. Each company with a network that connects to the Internet can be considered to be an autonomous system and can be assigned an ASN. (IANA assigns unique ASNs.) Each ISP has an ASN. Some large ISPs have multiple ASNs.

When a router uses BGP or BGP4+ to advertise a route, the prefix/ length is associated with a set of PAs, including the AS_Path. The AS_Path PA associated with a prefix/ length lists the ASNs that would be part of an end-to-end route for that prefix as learned using BGP or BGP4+.

BGP and BGP4+ use the AS_Path for two key functions:

- Choose the best route for a prefix based on the shortest AS_Path.
- Prevent routing loops.

Internal and External BGP Concepts

BGP defines two classes of neighbors (peers): internal BGP (iBGP) and external BGP (eBGP). These terms use the perspective of a single router, with the terms referring to whether a BGP neighbor is in the same ASN (iBGP) or a different ASN (eBGP).

A BGP router behaves differently in several ways depending on whether the peer (neighbor) is an iBGP or eBGP peer. The differences include different rules about what must be true before the two routers can become neighbors, different rules about which routes the BGP best path algorithm chooses as best, and even some different rules about how the routers update the BGP AS_Path PA.

When advertising to an eBGP peer, a BGP router updates the AS_Path PA, but it does not do so when advertising to an iBGP peer.

Public and Private ASNs

IANA administers the assignment of ASNs as it does with IP address prefixes. ASNs must be assigned as unique values since if ASNs are duplicated, the BGP loop prevention process may prevent parts of the Internet learning about a route.

IANA controls the ASN numbering space. Using the same process as for IPv4 addresses, ASNs are assigned to different organizations. The previous 16-bit BGP ASN has a decimal range of 0 through 65,535. The 32-bit BGP ASN has a decimal range of 1 through 4,294,967,295.

Like the public IPv4 address space, the public BGP ASN space has similar issues. To help overcome this issue, the ASN assignment process requires that each AS justify whether it truly needs a publicly unique ASN or whether it can just as easily use a private ASN. RFC 5398 reserves a small range of ASNs for use in documentation so that documentation avoids the use of ASNs assigned to organizations.

Private ASNs allow routers inside an AS to participate with BGP, using the same ASN as other organizations. An AS can use a private AS where the AS connects to only one other AS. Loops cannot occur at that point in the BGP topology, so unique ASNs in that network do not exist. Loops cannot occur with BGP advertising the best path for a prefix.

Outbound Routing Toward the Internet

The main reason to use BGP between an Enterprise and an ISP is to influence the choice of best path (or best route). However, since the majority of the end-to-end routes exist inside the Internet, it is hard to determine which exit point from the Enterprise is a better route.

Enterprises typically have two major classes of options for outbound routing toward the Internet: default routing and BGP. Using default routes is reasonable. This section discusses default routes toward the Internet, and outbound routes toward the Internet.

BGP is useful when you have at least two Internet connections. BGP is most useful when you want to choose one outbound path over another path for particular destinations in the Internet. Consider BGP when you have multiple Internet connections, where you want to make some packets take one path and you want other packets to take another path.

External BGP (eBGP) Concepts

This section describes the basics of eBGP to configure, verify, and advertise eBGP routes:

- **External BGP (eBGP)**: describes connections, configuration and commands to verify eBGP.
- **Verifying BGP**: describes the contents of the BGP table, and routes learned with eBGP.
- **Advertising eBGP routes to ISPs**: shows how to configure eBGP to advertise routes.

External BGP (eBGP)

BGP first forms a neighbor relationship with peers. BGP then learns information from its neighbors, placing information in a table named the BGP table. BGP analyzes the BGP table to choose the best route for each prefix in the BGP table, placing those routes into the IP routing table.

This section discusses the configuration of eBGP peers (called neighbors), with settings that may be needed for eBGP connections to result in working BGP neighborships, then discusses the BGP table, listing the learned prefix/ length and path attributes (PA).

eBGP Neighbor Configuration

At a minimum, a router participating in BGP needs to configure the following settings:

- The ASN for the router configured from a **router bgp (BGP and BGP4+)** command from Global Configuration mode on the router to enter Router Configuration mode.
- The IP address and ASN of each neighbor from a **neighbor remote-as (BGP and BGP4+)** command from Router Configuration mode, once the ASN is configured with the **router bgp (BGP and BGP4+)** command in the Global Configuration mode.

How Routers become eBGP Neighbors

Routers must meet several requirements to become BGP neighbors:

- A local router's ASN must match the reference from the neighboring router to that ASN as specified from the **router bgp (BGP and BGP4+)** command.
- The BGP router IDs of the two routers must not be the same.
- Each router must establish a TCP connection with the other router.

The remote router's IP address specified from the **neighbor remote-as (BGP and BGP4+)** command and used in that TCP connection must match what the local router configures from a **router bgp (BGP and BGP4+)** command.

How to verify eBGP Neighbor Status

The two most common commands to display a BGP neighbor's status are: **show ip bgp summary (BGP)** and **show ip bgp neighbors (BGP)**.

The **show ip bgp neighbors (BGP)** command shows detailed output per neighbor.

The **show ip bgp summary (BGP)** command shows summary output per neighbor.

Administratively Controlling Neighbor Status

To administratively disable any BGP neighbor, enter BGP configuration mode with the router **bgp** command and issue the **neighbor <neighborid> shutdown** command to set an idle state.

To administratively enable any BGP neighbor, enter BGP configuration mode with the router **bgp** command and issue the **no neighbor <neighborid> shutdown** command to set an idle state.

Use the **debug bgp (BGP)** command to see the BGP messages.

Verifying BGP

When a BGP router has established its eBGP neighbor relationships, that router can advertise and learn routes using BGP. To learn routes, a BGP router does not need additional configuration beyond the configuration of eBGP neighbor.

To advertise routes to eBGP peers, particularly the public IP address prefix(es) used by that Enterprise, the Enterprise BGP router needs some additional configuration.

The BGP table is important in the process of learning and using routing information with BGP. A router stores all learned BGP prefixes and PAs in its BGP table. The BGP router chooses which route for each prefix is the best BGP route. The BGP router advertises its BGP table to its neighbors, so the best route for each prefix is advertised to neighbors.

BGP Update Messages

When a BGP neighborhood reaches the established state, those neighbors begin sending BGP Update messages to each other. The router receiving an Update places those learned prefixes into its BGP table, regardless of whether the route appears to be the best route. BGP puts all learned routing information into its table, and then BGP processes all such potential routes to choose the best route for each prefix.

The format of the Update message tells us about BGP as a Path Vector algorithm. The message lists a set of PAs and then a long list of prefixes that use that set of PAs. So, you might view the BGP Update message as focusing on advertising paths, or a set of PAs, along with the associated list of prefixes that use the advertised path. BGP uses the information in the combined set of PAs to make a decision of which path is best.

Verify the BGP Table

Verify BGP to check the prefixes in the BGP table and confirm that the right prefixes have been learned from the expected neighbors. The BGP table should hold all learned prefixes, from each neighbor, except for any prefixes filtered by an inbound BGP filter.

For a router configured with a **neighbor route-map (BGP and BGP4+)** command, the local router would first filter the routes and then add the allowed routes into the BGP table. The **show ip bgp (BGP)** command lists the entire BGP routing table.

Viewing a subset of the BGP Table

When accepting full BGP updates, the number of BGP table entries may be too large for the **show ip bgp (BGP)** command listing thousands of prefixes. So instead use the **show ip bgp summary (BGP)** command that only lists the number of prefixes received from each neighbor.

Advertising eBGP routes to ISPs

Outbound routes let the Enterprise routers forward packets toward the Internet. At the same time, the ISPs need to learn routes for the Enterprise's public IP address space. This section of this chapter examines the options for advertising these routes. In particular, this section looks at two options:

- BGP **network (BGP and BGP4+)** command
- Redistribute from an IGP (RIP/RIPng/OSPF/OSPFv3)

Advertising eBGP routes with the network command

The BGP **network (BGP and BGP4+)** command installs the prefix defined in the BGP table to be advertised to peers. The route does not have to exist in the routing table. BGP does not use this command to enable BGP on interfaces. This command compares the command's parameters and the contents of that router's IP routing table, as follows:

Look for a route in the router's current IP routing table that exactly matches the parameters of the **network (BGP and BGP4+)** command. If a route for that exact prefix/length exists, then put the equivalent prefix/length into the local BGP table.

This assumes a BGP default setting of **no auto-summary**. With **auto-summary** configured, the router adds a route for that classful network to the BGP table:

- If the exact classful route is in the IP routing table
- If any subset routes of that classful network are in the routing table

Advertising eBGP routes with the redistribute command

Instead of using the **network (BGP and BGP4+)** command to add routes to the BGP table, BGP routers can use a **redistribute** command for OSPF, OSPFv3, RIPng, and RIP to redistribute routes from an IGP into BGP.

- Advertise the public address range, not the private IP address range, to the BGP table.
- Advertise one route for the public address range, not individual subnets of the range.

Routers that run BGP may already run an IGP and have learned routes for either the entire public range either as one route or with subset routes. If a single route exists for the entire public range, then you need to add a **redistribute** command for OSPF, OSPFv3, RIPng, and RIP to the BGP configuration to redistribute only that single route into BGP.

Internal BGP (iBGP) Concepts

Routers that run BGP often run an IGP, such as OSPF, and have learned routes for the public range either as one route or with subset routes. If a single route exists for the public range, then you can add a **redistribute (into BGP or BGP4+) (BGP and BGP4+)** command to the BGP configuration to redistribute that route, and only that route, into BGP.

IGPs do not handle the public routing table. Internally peered BGP routers are capable of exchanging routing information without redistributing into an IGP. But an IGP can be used to connect two or more distant iBGP peers, and redistribution into an IGP may be used.

iBGP between Internet connected routers

When an organization uses more than one router to connect to the Internet, and those routers use BGP to exchange routing information with their ISPs, those same routers need to exchange BGP routes with each other as well. The BGP neighbor relationships occur inside that organization, making these routers iBGP peers.

iBGP with two Internet connected routers

Two Internet-connected routers in an Enterprise need to communicate BGP routes to each other because these routers may want to forward IP packets to the other Internet-connected router, which in turn would forward the packets into the Internet.

With an iBGP peer connection, each Internet-connected router can learn routes from the other router and decide if that other router has a better route to reach some destinations in the Internet. Without that iBGP connection, the routers have no way to know if the other router has a better BGP path.

Configuring iBGP

The most basic iBGP configuration differs only slightly compared to eBGP configuration. The configuration does not explicitly identify an eBGP versus an iBGP peer. Instead, for iBGP, the neighbor's ASN listed on the **neighbor remote-as (BGP and BGP4+)** command lists the same ASN as the local router's **router bgp (BGP and BGP4+)** command. This is because the local router and the neighbor are in the same ASN.

For eBGP, the **neighbor remote-as (BGP and BGP4+)** command lists a different ASN.

The configuration to use loopback interfaces as the update source mirrors configuration for eBGP peers, except that iBGP peers do not need to configure the **neighbor ebgp-multihop (BGP and BGP4+)** command.

For iBGP connections, only the following steps are required to make two iBGP peers use a loopback interface:

1. Configure an IP address on a loopback interface on each router.
2. Configure each router to use the loopback IP address as the source IP address, for the neighborhood with the other router, using the **neighbor update-source (BGP and BGP4+)** command.
3. Configure the BGP **neighbor remote-as (BGP and BGP4+)** command on each router to refer to the other router's loopback IP address as the neighbor IP address in the **neighbor remote-as (BGP and BGP4+)** command.

4. Make sure each router has IP routes so that they can forward packets to the loopback interface IP address of the other router.

Verifying iBGP

iBGP neighbors use the same messages and neighbor states as eBGP peers. The same commands for BGP neighbor verification can be used to verify iBGP peers. The **show ip bgp summary (BGP)** command lists neighbors. The **show ip bgp neighbors (BGP)** and **show bgp ipv6 neighbors (BGP4+)** commands list many details specifically for the neighbor for BGP and BGP4+ respectively.

Understanding Next-Hop Reachability Issues with iBGP

With IGP, the IP routes added to the IP routing table list a next-hop IP address. With few exceptions, the next-hop IP address routes exist in a connected subnet.

BGP advertises these particular IP addresses as the next-hop IP addresses because of a default behavior for BGP. By default, when a router advertises a route using eBGP, the advertising router lists its own update-source IP address as the next-hop address of the route. In other words, the next-hop IP address is the IP address of the eBGP neighbor, as listed on the **neighbor remote-as (BGP and BGP4+)** command. However, when advertising a route to an iBGP peer, the advertising router (by default) does not change the next-hop address.

The IP routing process can use routes whose next-hop addresses are not in connected subnets as long as each router has an IP route that matches the next-hop IP address. Two main options exist to ensure reachability to these next-hop addresses:

- Create IP routes so that each router can reach these next-hop addresses that exist in other ASNs.
- Change the default iBGP behavior with the **neighbor next-hop-self (BGP and BGP4+)** command.

Using neighbor next-hop-self to change Next-Hop Address

The second option for dealing with these unconnected next-hop IP addresses changes the iBGP configuration, so that a router changes the next-hop IP address on iBGP-advertised routes. This option requires the **neighbor next-hop-self (BGP and BGP4+)** command to be configured for the iBGP neighbor relationship.

With this command configured, the router advertises iBGP routes with its own update-source IP address as the next-hop IP address. Because the iBGP neighborship already relies on a working route for these update source IP addresses, if the neighborship is up, then IP routes already exist for these next-hop addresses.

IGP Redistribution and BGP Synchronization

Redistributing BGP routes into the IGP can solve routing loop problems. Redistributing BGP routes into the IGP prevents routing loops by giving routers the best exit point for each destination.

Redistributing eBGP-learned Internet routes into the IGP is the BGP feature started with the **synchronization (BGP and BGP4+)** command, where iBGP learned routes must be synchronized with IGP-learned routes for the same prefix before they can be used. If an iBGP learned route is to be considered to be a usable route, then that same prefix must be in the IP routing table and learned locally or via an IGP, such as OSPF.

The synchronization feature tells a BGP router not to consider an iBGP-learned route as best unless the prefix is learned locally or via an IGP and is currently in the IP routing table.

BGP synchronization prevents reachability issues by preventing a BGP router advertising routes learned from an iBGP peer unless the route is learned locally, or via an IGP.

If the route does not exist in the routing table, because an IGP has not learned the route, then BGP will not use or advertise the learned prefix. If BGP advertised the route to an external peer, then the external peer would consider that path valid. This is the resulting problem if synchronization is not enabled and there are non-BGP routers in the transit AS.

The eBGP peer would send traffic to the BGP router advertising the prefix and that router would be able to pass the traffic on since the IGP routers between the two iBGP routers do not know the prefix.

How to configure BGP and BGP4+

This section describes BGP and BGP4+ configuration tasks and the commands required. For example configurations with sample topologies, see [Chapter 2, BGP and BGP4+ Configuration](#).

Initial BGP and BGP4+ Configuration Tasks

There are two initial BGP and BGP4+ configuration tasks described in the below sections. More advanced BGP and BGP4+ configuration tasks are described later in this chapter and are also available separately in the How to Configure BGP and BGP4+ Technical Guide:

- [How to enable BGP and BGP4+ Routing](#)
- [How to configure BGP and BGP4+ Neighbors](#)

How to enable BGP and BGP4+ Routing

To enable BGP or BGP4+ routing and establish a BGP or BGP4+ routing process, use the following commands starting in Global Configuration mode:

```
awplus(config)#
router bgp <asn>
```

This command enables a BGP or BGP4+ routing process, which places the router in Router Configuration mode.

```
awplus(config-router)#
network {<ip-prefix/length> |
  <ip-network-addr>} [mask
  <network-mask>] [route-map
  <route-map-name>] [backdoor]
```

This command flags an IPv4 network as local to this autonomous system and enters it to the BGP table.

```
awplus(config-router)#
network {<ipv6-prefix/length> |
  <ipv6-network-addr>} [route-map
  <route-map-name>]
```

This command flags an IPv6 network as local to this autonomous system and enters it to the BGP4+ table.

How to configure BGP and BGP4+ Neighbors

Like other EGP, BGP must completely understand the relationships it has with its neighbors. Therefore, this task is required.

BGP supports two kinds of neighbors: internal and external. Internal neighbors are in the same autonomous system; external neighbors are in different autonomous systems. Normally, external neighbors are adjacent to each other and share a subnet, while internal neighbors may be anywhere in the same autonomous system.

To configure BGP neighbors, use the following command in Router Configuration mode:

```
awplus(config-router)#
neighbor <neighborid> remote-as
  <as-number>
```

This command specifies a BGP neighbor.

BGP and BGP4+ Neighbor Route Filtering

You can filter BGP and BGP4+ advertisements in two ways with neighbors:

- Use autonomous system path filters, as with the **ip as-path access-list (BGP and BGP4+)** Global Configuration mode command and the **neighbor filter-list (BGP and BGP4+)** Router Configuration mode command
- Use access or prefix lists, as with the **neighbor distribute-list (BGP and BGP4+)** Router Configuration mode command.

If you want to restrict routing information that is learned or advertised, you can filter BGP routing updates to and from neighbors. You can define an access list or a prefix list and apply it to the updates.

To filter BGP routing updates for IPv4 routes, use the command below in Router Configuration mode:

```
awplus(config-router)#
neighbor <neighborid> This command filters BGP routing updates to and from
distribute-list <access-list> neighbors as specified in an access list for IPv4 routers in
                        {in|out} Router Configuration mode.
```

To filter BGP4+ routing updates for IPv6 routes, use the command below in Address Family Configuration mode:

```
awplus(config-router-af)#
neighbor <neighborid> This command filters BGP routing updates to and from
distribute-list <access-list> neighbors as specified in an access list for IPv6 routes in
                        {in|out} Address Family Configuration mode.
```

BGP and BGP4+ Path Filtering by Neighbor

You can specify an access list filter on both incoming and outbound updates based on the BGP autonomous system paths. Each filter is an access list based on regular expressions. To specify the access list filter, define an autonomous system path access list and apply it to updates to and from particular neighbors.

To configure BGP path filtering, use the following commands starting in Global Configuration mode:

```
awplus(config)#
ip as-path access-list (BGP and This command defines a BGP-related access list.
                        BGP4+)
```

```
awplus(config)#
router bgp <asn> This command enters Router configuration mode
```

```
awplus(config-router)#
neighbor <neighborid> This command establishes a BGP filter for an IPv4 path in the
filter-list <listname> {in|out} Router Configuration mode.
```

To configure BGP4+ path filtering, use the following commands starting in Global Configuration mode:

<code>awplus(config)#</code>	
<code>ip as-path access-list (BGP and BGP4+)</code>	This command defines a BGP-related access list.

<code>awplus(config)#</code>	
<code>router bgp <asn></code>	This command enters Router Configuration mode

<code>awplus(config-router)#</code>	
<code>address-family ipv6 [unicast]</code>	This command enters Address Family Configuration mode

<code>awplus(config-router-af)#</code>	
<code>neighbor <neighborid></code>	This command establishes a BGP4+ filter for an IPv6 path in the Address Family Configuration mode.
<code>filter-list <listname> {in out}</code>	

How to disable next hop processing for BGP or BGP4+ updates

You can configure the software to disable next hop processing for BGP and BGP4+ updates to a neighbor.

For BGP, to disable next hop processing and provide a specific address to be used instead of the next hop address, use the command below in Router Configuration mode:

<code>awplus(config-router)#</code>	
<code>neighbor <neighborid></code>	This command disables next hop processing on BGP updates to an IPv4 neighbor in Router Configuration mode.
<code>next-hop-self</code>	

For BGP4+, to disable next hop processing and provide a specific address to be used instead of the next hop address, use the command below in Address Family Configuration mode:

<code>awplus(config-router-af)#</code>	
<code>neighbor <neighborid></code>	This command disables next hop processing on BGP4+ updates to an IPv6 neighbor in Address Family Configuration mode.
<code>next-hop-self</code>	

Configuring this command causes the current router to advertise its peering address as the next hop for the specified neighbor. Therefore, other BGP neighbors will forward to it packets for that address.

How to use Route Maps for BGP or BGP4+ updates

You can use a route map on a per-neighbor basis to filter updates and modify various attributes. A route map can be applied to either inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates.

Note that community based matching requires the **ip community-list (BGP and BGP4+)** Global Configuration command.

See the section **BGP Community Filtering** for more information. See also the related route-map commands that are included in the **BGP and BGP4+ Commands** chapter of this Supplement: **set community (Route Map)** and **match community (Route Map)**.

For BGP, to apply a route map to incoming and outgoing IPv4 routes, use the command below in the Router Configuration mode:

```
awplus(config-router)#  
neighbor <neighborid> route-map <mapname> {in|out}
```

This command applies a route map to incoming or outgoing IPv4 routes in the Router Configuration mode.

For BGP4+, to apply a route map to incoming and outgoing IPv6 routes, use the following command in Address Family Configuration mode:

```
awplus(config-router-af)#  
neighbor <neighborid> route-map <mapname> {in|out}
```

This command applies a route map to incoming or outgoing IPv6 routes in the Address Family Configuration mode.

BGP and BGP4+ Peer Groups

Often, in a BGP speaker, many neighbors are configured with the same update policies. Neighbors with the same update policies can be grouped into peer groups to simplify configuration and, more importantly, to make updating more efficient.

Three steps to configure a BGP peer group, described in following sections, are as below:

1. **How to create a Peer Group for BGP**
2. **How to assign options to the created Peer Group**
3. **How to make neighbors members of a Peer Group**

You can disable a BGP peer or peer group without removing all the configuration information using the **neighbor shutdown (BGP and BGP4+)** Router Configuration mode command.

Note the steps differ slightly for BGP4+ peer groups. First you create a peer group in Global Configuration mode. Next you activate the peer group in Address Family Configuration mode, not Router Configuration mode. Then you assign options to the created peer group in Address Family Configuration mode, not Router Configuration mode as used for BGP.

How to create a Peer Group for BGP

To create a BGP peer group, use the following command in Router Configuration mode:

```
awplus(config-router)#  
neighbor peer-group (create a peer-group) (BGP and BGP4+)
```

This command creates a BGP peer group.

How to assign options to the created Peer Group

After you create a peer group, you configure the peer group with neighbor commands. By default, members of the peer group inherit all the configuration options of the peer group. Members can also be configured to override the options that do not affect outbound updates.

Peer group members will always inherit the following attributes: minimum-advertisement-interval, next-hop-self, out-route-map, out-filter-list, out-dist-list, remote-as (if configured), version, and update-source. All the peer group members will inherit changes made to the peer group.

To assign configuration options to an individual neighbor, specify any of the following commands using the IP address as the `<neighborid>`. To assign the options to a peer group, specify any of the commands using the peer group name as the `<neighborid>`. Use the following commands in Router Configuration mode as needed.

<pre>awplus(config-router)# neighbor <neighborid> remote-as <as-number></pre>	This command specifies a BGP neighbor.
<pre>awplus(config-router)# neighbor <neighborid> description <description></pre>	This command associates a description with a neighbor.
<pre>awplus(config-router)# neighbor {<neighborid>} default-originate [route-map <routemap-name>]</pre>	This command allows a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route.
<pre>awplus(config-router)# neighbor <neighborid> send-community {both extended standard}</pre>	This command specifies that the community attribute be sent to the neighbor at this IP address.
<pre>awplus(config-router)# neighbor <neighborid> update-source <interface></pre>	This command allows iBGP sessions to use any operational interface for TCP connections.
<pre>awplus(config-router)# neighbor <neighborid> ebgp-multihop [<count>]</pre>	This command allows BGP sessions, even when the neighbor is not on a directly connected segment. The multihop session is not established if the only route to the address of the multihop peer is the default route (0.0.0.0).
<pre>awplus(config-router)# neighbor <neighborid> advertisement-interval <time></pre>	This command sets the minimum interval between sending BGP routing updates.
<pre>awplus(config-router)# neighbor <neighborid> maximum-prefix <maximum></pre>	This command limits the number of prefixes allowed from a neighbor.
<pre>awplus(config-router)# neighbor <neighborid> weight <weight></pre>	This command specifies a weight for all routes from a neighbor.
<pre>awplus(config-router)# neighbor <neighborid> distribute-list <access-list> {in out}</pre>	This command filters BGP routing updates to and from neighbors, as specified in an access list..
<pre>awplus(config-router)# neighbor <neighborid> filter-list <listname> {in out}</pre>	This command establishes a BGP filter.

<pre>awplus(config-router)# neighbor <neighborid> next-hop-self</pre>	This command disables next hop processing on the BGP updates to a neighbor.
<pre>awplus(config-router)# neighbor <neighborid> version <version></pre>	This command specifies the BGP version to use when communicating with a neighbor.
<pre>awplus(config-router)# neighbor {<ip-address> <peer-group-name>} password <password></pre>	This command starts MD5 authentication on a TCP connection to a BGP peer. You can enter a case-sensitive password of up to 25 characters. The string can contain any alphanumeric characters.
<pre>awplus(config-router)# neighbor <neighborid> route-map <mapname> {in out}</pre>	This command applies a route map to incoming or outgoing routes.
<pre>awplus(config-router)# neighbor <neighborid> soft-reconfiguration inbound</pre>	This command configures the software to start storing received updates.

If a peer group is not configured with a remote-as attribute, the members can be configured with the **neighbor remote-as (BGP and BGP4+)** command in Router Configuration mode. This command allows you to create peer groups containing external BGP (eBGP) neighbors.

You can customize inbound policies for peer group members (using a distribute list, a route map, or a filter list) because one identical copy of an update is sent to every member of a group. Neighbor options for outgoing updates cannot be customized for peer groups.

External BGP (eBGP) peers must reside on a directly connected network. But you may want to change this to test BGP or BGP4+. You can do this by specifying the **neighbor ebgp-multihop (BGP and BGP4+)** command in Router Configuration mode. Members of a peer group can pass routes from one member of the peer group to another. For example, if router B is peering with routers A and C, router B can pass routes from router A to router C.

For Internal BGP (iBGP) peers, you may want to allow BGP or BGP4+ connections to stay up if any interface is used. To enable this configuration, configure a loopback interface and assign it an IPv4 or an IPv6 address. Next, configure the BGP or BGP4+ update source to be the loopback interface. Then configure your neighbor to use the address on the loopback interface. Now the iBGP session will be up as long as there is a route on any interface.

How to configure authentication for BGP Peering Sessions

You can configure authentication between two BGP peers, meaning that each segment sent on the TCP connection between the peers is verified. Authentication must be configured with the same password on both BGP peers; otherwise, the connection between them will not be made. Configuring authentication causes the software to generate and check the MD5 digest of every segment sent on the TCP connection. If authentication is invoked and a segment fails authentication, then an error message will be displayed in the console.

```
awplus(config-router)#  
neighbor {<ip-address> | <peer-group-name>} password <password>
```

This command starts MD5 authentication on a TCP connection to a BGP peer. You can enter a case-sensitive password of up to 25 characters.

When configuring authentication, you can enter a case-sensitive password of up to 25 characters. The string can contain any alphanumeric characters, including spaces. A password cannot be configured in the number-space-anything format. The space after the number can cause authentication to fail.

How to make neighbors members of a Peer Group

To configure a BGP neighbor to be a member of a BGP peer group, use the following command in Router Configuration mode, using the same peer group name:

```
awplus(config-router)#  
neighbor <ipv6-addr> peer-group <peer-group>
```

This command makes a BGP neighbor a member of the peer group.

How to disable a Peer or Peer Group

To disable an existing BGP neighbor or neighbor peer group, use the following command in Router Configuration mode:

```
awplus(config-router)#  
neighbor <neighborid> shutdown
```

This command shuts down or disables a BGP neighbor or peer group.

To enable a previously existing neighbor or neighbor peer group that had been disabled using the **neighbor shutdown (BGP and BGP4+)** command in Router Configuration mode, use the following command in Router Configuration mode:

```
awplus(config-router)#  
no neighbor <neighborid> shutdown
```

This command enables a BGP neighbor or peer group.

BGP and BGP4+ Prefix Lists

Prefix lists can be used as an alternative to access lists in many BGP or BGP4+ route filtering commands. The advantages of using IPv4 or IPv6 prefix lists are as follows:

- More of a performance improvement in loading and route lookup of large lists.
- Support for updates. Filtering using access lists does not support updates.
- The command-line interface to use access lists to filter BGP updates is more difficult.
- More flexibility

Before using a prefix list in a BGP or BGP4+ command, first set up the prefix list, and assign sequence numbers to entries in the prefix list.

BGP and BGP4+ filtering by prefix lists involves matching the prefixes of IPv4 or IPv6 routes with those listed in the prefix list. When there is a match, the route is used. Whether a prefix is permitted or denied is based upon these rules:

- An empty prefix list permits all prefixes.
- An implicit deny is assumed if a given prefix does not match any entries of a prefix list.

The router begins the search at the top of the prefix list, with the sequence number 1. Once a match or deny occurs, the router need not go through the rest of the prefix list.

For efficiency, add the most common matches or denies near the top of the list, using the **seq** parameter in the **ip prefix-list (IPv4 Prefix List)** and **ipv6 prefix-list (IPv6 Prefix List)** Global Configuration mode commands. The **show ip prefix-list (IPv4 Prefix List)** command and the **show ipv6 prefix-list (IPv6 Prefix List)** command output includes sequence numbers.

Sequence numbers are generated automatically unless you disable automatic generation. If you disable the automatic generation of sequence numbers, you need to specify the sequence number for each entry with the **seq** parameter of the **ip prefix-list (IPv4 Prefix List)** and **ipv6 prefix-list (IPv6 Prefix List)** Global Configuration mode commands.

How to create and configure a Prefix List for BGP or BGP4+

To create a prefix list for BGP, use the following command in Global Configuration mode. Note you can add individual prefix list entries to a defined prefix list as well.

```
awplus(config)#
ip prefix-list <list-name>
[seq <1-429496725>] {deny|
permit} {any|<ip-prefix>} [ge
<0-32>] [le <0-32>]
```

This command creates an IPv4 prefix list with the name specified for the <list-name> parameter. To create a prefix list you must enter at least one **permit** or **deny** clause. Once an IPv4 prefix list is created, this command creates a prefix list entry, and assigns a sequence number to the IPv4 prefix list entry.

```
awplus(config)#
ipv6 prefix-list <list-name>
[seq <1-429496725>] {deny|
permit} {any|<ipv6-prefix>} [ge
<0-128>] [le <0-128>]
```

This command creates an IPv6 prefix list with the name specified for the <list-name> parameter. To create a prefix list you must enter at least one **permit** or **deny** clause. Once an IPv6 prefix list is created, this command creates a prefix list entry and assigns a sequence number to the IPv6 prefix list entry.

The optional **ge** and **le** keywords can be used to specify the range of the prefix length to be matched for prefixes that are more specific than the network/length parameter. An exact match is assumed when neither **ge** nor **le** is specified.

You can specify sequence values for prefix list entries in any increments you want. If you specify the sequence values in increments of 1, you cannot insert additional entries into the prefix list. If you choose very large increments, you could run out of sequence values.

How to delete Prefix List Entries

To remove a prefix list, and optionally specific entries, use the following commands as appropriate in Global Configuration mode:

```
awplus(config)#
no ip prefix-list <list-name>
[seq <1-429496725>]
```

This command removes an IPv4 prefix list with the name specified for the required <list-name> parameter and the sequence number specified in the range <1-429496725>.

```
awplus(config)#
no ipv6 prefix-list <list-name>
[seq <1-429496725>]
```

This command removes an IPv6 prefix list with the name specified for the <list-name> parameter and the sequence number specified in the range <1-429496725>.

The sequence number of an entry is optional when you delete the prefix list entry. If you omit the sequence number then all entries are deleted for the prefix list.

How to display IPv4 and IPv6 Prefix Entries

To display information about IPv4 and IPv6 prefix tables, prefix table entries, the policy associated with a node, or specific information about an entry, use the following commands in Privileged Exec mode as appropriate:

```
awplus#  
show ip prefix-list [<name> |  
  detail | summary]
```

This command displays information about IPv4 prefix lists.

```
awplus#  
show ipv6 prefix-list [<name> |  
  detail | summary]
```

This command displays information about IPv6 prefix lists.

BGP and BGP4+ Route Dampening

Route dampening is a BGP and BGP4+ feature designed to minimize the propagation of flapping routes across a network. A route is considered to be flapping when it is repeatedly available, then unavailable, then available, then unavailable, etc.

Consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. If the route to network A in autonomous system 1 flaps, and is unavailable without route dampening, the eBGP neighbor of autonomous system 2 sends a withdraw message to autonomous system 2.

The router in autonomous system 2 sends the withdraw message to autonomous system 3. When a route to network A appears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3.

If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in a network connected to the Internet since route flapping in the Internet often involves many routes.

The route dampening feature minimizes the flapping problem as follows. If the route to network A flaps, then the router in autonomous system 2, with route dampening enabled, assigns network A a penalty of 1000 and moves it to the history state.

The router in autonomous system 2 continues to advertise the status of the route to neighbors. When the route flaps so the penalty exceeds the limit, the router stops advertising the route to network A, so the route becomes dampened.

The penalty placed on network A is reduced until the limit is reached, when the route is advertised. At half the limit, dampening information for a route to network A is removed.

The below terms are used for BGP and BGP4+ route dampening:

- **Flap**—A route is available, then unavailable, or vice versa.
- **History state**—After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- **Penalty**—Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
- **Damp state**—In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- **Suppress limit**—A route is suppressed when its penalty exceeds this limit. The default value is 2000.
- **Half-life**—Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.

How to enable BGP and BGP4+ Route Dampening

To enable BGP route dampening, use the command below in Router Configuration mode:

```
awplus(config)#
router bgp <asn>
```

This command enters Router Configuration mode

```
awplus(config-router)#
  bgp dampening
```

This command enables BGP route dampening in Router Configuration mode.

To enable BGP4+ route dampening, use the command below in Address Family Configuration mode:

```
awplus(config)#
router bgp <asn>
```

This command enters Router Configuration mode

```
awplus(config-router)#
address-family ipv6 [unicast]
```

This command enters Address Family Configuration mode

```
awplus(config-router-af)#
  bgp dampening
```

This command enables BGP4+ route dampening in Address Family Configuration mode.

How to monitor BGP and BGP4+ Route Dampening

You can monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life. To display flap statistics, use the following show commands in Privileged Exec mode as needed:

```
awplus#
show ip bgp dampening
  {dampened-paths |
  flap-statistics|parameters}
```

This command displays BGP flap statistics for all IPv4 paths when you apply the **flap-statistics** parameter.

```
awplus#
show bgp ipv6 dampening
  {dampened-paths |
  flap-statistics|parameters}
```

This command displays BGP4+ flap statistics for all IPv6 paths when you apply the **flap-statistics** parameter.

Once a route is dampened, you can display BGP route dampening information, including the time remaining before the dampened routes will be unsuppressed. To display the information, use the following command in Privileged Exec mode:

```

awplus#
show ip bgp dampening {dampened-paths | flap-statistics|parameters}

```

This command displays BGP route dampening information for all IPv4 paths when you apply the **dampened-paths** parameter.

```

awplus#
show bgp ipv6 dampening {dampened-paths | flap-statistics|parameters}

```

This command displays BGP4+ route dampening information for all IPv6 paths when you apply the **dampened-paths** parameter.

To clear BGP flap statistics (thus making it less likely that the IPv4 route will be dampened), use the following command in Privileged Exec mode :

```

awplus#
clear ip bgp flap-statistics [<ip-address>|<ip-address/m>]

```

This command clears BGP flap statistics for all IPv4 routes.

To clear BGP dampening statistics for IPv4 routes, use the following command in Privileged Exec mode:

```

awplus#
clear ip bgp dampening [<ip-address>|<ip-address/m>]

```

This command clears BGP dampening statistics for all IPv4 routes.

To clear BGP4+ flap statistics (thus making it less likely that the IPv6 route will be dampened), use the following command in Privileged Exec mode :

```

awplus#
clear bgp ipv6 flap-statistics [<ipv6-addr>|<ipv6-addr/prefix-length>]

```

This command clears BGP4+ flap statistics for all IPv6 routes.

To clear BGP4+ dampening statistics for IPv6 routes, use the following command in Privileged Exec mode:

```

awplus#
clear bgp ipv6 dampening [<ipv6-addr>|<ipv6-addr/prefix-length>]

```

This command clears BGP4+ dampening statistics for all IPv6 routes.

The flap statistics for a route are also cleared when a BGP or BGP4+ peer is reset. Although the reset withdraws the route, there is no penalty applied, even if route flap dampening is enabled.

BGP and BGP4+ Synchronization

IGP synchronisation is a feature that is specifically required in the case where routing information is 'transiting' through an AS. That is, when:

- A BGP router on one side of the AS learns routes from external ASs by eBGP.
- This router passes the routes across the AS by iBGP to a BGP router on the other side of the network (passing through several IGP routers on the way, which simply pass on the iBGP updates).
- The BGP router on the other side of the network then sends the routes by eBGP to other external ASs.

The problem that can arise in this situation comes about because the iBGP session effectively tunnels route information across the AS, but packets destined to any of those routes have to cross the AS hop by hop.

If iBGP tunnels route information across the AS faster than the AS's IGP can pass the route information to all routers in between, then we end up with a situation where the BGP routers (and other ASs) know about routes that some of the IGP routers in the middle of the transit AS have not learnt yet. So, packets that are routed from an external AS into the transit AS may end up being dropped or misdirected somewhere in the middle of the AS when they arrive at an IGP router that has not yet learnt all the transiting routes.

To avoid this problem, we need to tell iBGP routers that even if they receive routes via the iBGP 'express tunnel', they must wait until the routes have also been received via the more pedestrian iBGP hop-by-hop method before the routes can be passed on to eBGP neighbors in other ASs.

This process of forcing a router to wait until a route has been learnt by IGP as well as iBGP is called IGP synchronisation. In other words, the routes that the router has learnt by iBGP must match up with the routes that it has learnt by IGP. Once a route has achieved this sync-up, it is then OK to advertise it on to eBGP neighbors.

IGP synchronization is disabled by default. It does not need to be enabled in an AS where there are no iBGP connections, or in an AS where all the internal route advertising is done by iBGP (i.e., when there is no other IGP in use in the AS).

To enable BGP synchronization for IPv4 routes, use the command below in the Router Configuration mode:

```
awplus(config-router)#  
    synchronization
```

 This command enables synchronization between BGP and an IGP for IPv4 routes in Router Configuration mode.

To enable BGP4+ synchronization for IPv6 routes, use the command below in the Address Family Configuration mode:

```
awplus(config-router-af)#  
    synchronization
```

 This command enables synchronization between BGP4+ and an IGP for IPv6 routes in Address Family Configuration mode.

BGP and BGP4+ Weights

A weight is a number that you can assign to a route so that you can control the route selection process. The administrative weight is local to the router. A weight can be a number from 0 to 65535. If you have particular neighbors that you want to prefer for most of your traffic, you can assign a higher weight to all routes learned from that neighbor.

To assign a weight to a BGP route, use the following commands in Router Configuration mode:

```
awplus(config)#  
router bgp <asn>
```

 This command enters Router Configuration mode

```
awplus(router-config)#  
neighbor <neighborid> weight  
    <weight>
```

 This command adds a weight to a BGP route specified in the required <neighborid> parameter.

To assign a weight to a BGP4+ route, use the following commands in Address Family Configuration mode:

```
awplus(config)#  
router bgp <asn>
```

 This command enters Router Configuration mode

```
awplus(config-router)#  
address-family ipv6 [unicast]
```

 This command enters Address Family Configuration mode

```
awplus(router-config-af)#  
neighbor <neighborid> weight  
    <weight>
```

 This command adds a weight to a BGP4+ route specified in the required <neighborid> parameter.

BGP and BGP4+ Aggregate Addresses

CIDR (Classless Inter-Domain Routing) enables you to create aggregate routes to minimize the size of routing tables. CIDR is a routing method supported by BGP and BGP4+ that is based on route aggregation. CIDR allows a router to group routes together to minimize the amount of routing information carried. Several IP networks appear as a single network to networks outside of those grouped together.

You can configure aggregate routes in BGP or BGP4+ either by redistributing an aggregate route into BGP or BGP4+ or by using the BGP and BGP4+ Aggregation feature. An aggregate address is added to the BGP table if at least one entry is in the BGP table.

To create an IPv4 aggregate address in the BGP routing table, use the following command in the Router Configuration mode:

```
awplus(config-router)#
aggregate-address <ip-addr/m> {summary-only|as-set}
```

This command creates an aggregate IPv4 route entry in the BGP routing table. This entry will cause BGP to advertise IPv4 summary addresses only, if the **summary-only** parameter is applied. This entry will cause BGP to generate autonomous system set path information, if the the **as-set** parameter is applied.

To create an IPv6 aggregate address in the BGP4+ routing table, use the following command in the Address Family Configuration mode:

```
awplus(config-router-af)#
aggregate-address <ipv6-addr/
prefix-length> {summary-only|
as-set}
```

This command creates an aggregate IPv6 route entry in the BGP4+ routing table. This entry will cause BGP4+ to advertise IPv6 summary addresses only, if the **summary-only** parameter is applied. This entry will cause BGP4+ to generate autonomous system set path information, if the the **as-set** parameter is applied.

How to enable and disable Automatic Summarization

When a subnet is redistributed from an IGP to BGP, network routes are added to routing tables. By default, automatic summarization is disabled. To enable automatic network number summarization, use the following command in Router Configuration mode:

```
awplus(config-router)#
auto-summary
```

This command enables automatic network summarization.

To disable automatic network number summarization, use the following command in Router Configuration mode:

```
awplus(config-router)#
no auto-summary
```

This command disables automatic network summarization.

BGP and BGP4+ monitoring

You can remove all contents of a particular cache, table, or database. You also can display specific statistics. The following sections describe each of these tasks.

How to clear BGP and BGP4+ tables

You can remove all contents of a particular table. Clearing a table may be needed when the contents of the table have become invalid. To clear tables for BGP and BGP4+, use the following commands in Privileged Exec mode, as needed:

```
awplus#
clear ip bgp <ipv4-addr> This command resets a specified BGP IPv4 connection.
[in [prefix-filter]|out|soft
[in|out]]
```

This command resets all BGP IPv4 connections by clearing all BGP peers:

```
awplus#
clear ip bgp * This command resets all BGP IPv4 connections.
```

This command resets all BGP IPv4 and BGP4+ IPv6 connections by clearing all BGP and BGP4+ peers:

```
awplus#
clear bgp * This command resets all BGP IPv4 and BGP4+ IPv6 connections.
```

These commands remove all members of a BGP or BGP4+ peer group respectively:

```
awplus#
clear bgp peer-group <peer-group> This command removes all members of a BGP peer group.
[in [prefix-filter]|out|soft
[in|out]]
```

```
awplus#
clear ip bgp peer-group <peer- This command removes all members of a BGP peer group.
name>
```

```
awplus#
clear bgp ipv6 peer-group <peer- This command removes all members of a BGP4+ peer group.
name>
```

How to display BGP and BGP4+ statistics

You can display specific statistics such as the contents of BGP routing tables.

Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that the packets of your device are taking through the network.

To display various routing statistics, use the following commands in Privileged Exec mode, as needed:

awplus# show ip bgp prefix-list <list>	This command displays peer groups and peers not in peer groups to which the prefix has been advertised for BGP. It also displays prefix attributes, like next hop and local prefix.
awplus# show bgp ipv6 prefix-list <list>	This command displays peer groups and peers not in peer groups to which the prefix has been advertised for BGP4+. It also displays prefix attributes, like next hop and local prefix.
awplus# show ip bgp cidr-only	This command displays all BGP routes that contain subnet and supernet network masks.
awplus# show ip bgp community [<type>] [exact-match]	This command displays routes that belong to the specified communities for BGP.
awplus# show ip bgp community-list <listname> [exact-match]	This command displays routes that are permitted by the community list for BGP.
awplus# show bgp ipv6 community [<type>] [exact-match]	This command displays routes that belong to the specified communities for BGP4+.
awplus# show bgp ipv6 community-list <listname> [exact-match]	This command displays routes that are permitted by the community list for BGP4+.
awplus# show ip bgp filter-list <listname>	This command displays BGP IPv4 routes that are matched by the specified autonomous system path access list.
awplus# show bgp ipv6 filter-list <listname>	This command displays BGP4+ IPv6 routes that are matched by the specified autonomous system path access list.
awplus# show ip bgp inconsistent-as	This command displays the BGP IPv4 routes with inconsistent originating autonomous systems.
awplus# show bgp ipv6 inconsistent-as	This command displays the BGP4+ IPv6 routes with inconsistent originating autonomous systems.

<pre>awplus# show ip bgp regexp <expression></pre>	<p>This command displays the routes that have an autonomous system path that matches the specified regular expression entered on the command line.</p>
<pre>awplus# show ip bgp [<ip-addr> <ip-addr/m>]</pre>	<p>This command displays the contents of the IPv4 routes in the BGP routing table.</p>
<pre>awplus# show bgp ipv6 <ipv6-addr></pre>	<p>This command displays the contents of the IPv6 routes in the BGP4+ routing table.</p>
<pre>awplus# show ip bgp neighbors [<ipv4-addr> [advertised-routes received prefix-filter received-routes routes]]</pre>	<p>This command displays detailed information on connections to individual BGP neighbors. Note this command does not display IPv4 routes learned from a specified BGP neighbor. It shows BGP session information and the number of announced and accepted prefixes. A counter shows how many routes have been announced and accepted, but not learned.</p>
<pre>awplus# show bgp ipv6 neighbors [<ipv6- addr> [advertised-routes received prefix-filter received-routes routes]]</pre>	<p>This command displays detailed information on connections to individual BGP4+ neighbors. Note this command does not display IPv6 routes learned from a specified BGP4+ neighbor. It shows BGP4+ session information and the number of announced and accepted prefixes. A counter shows how many routes have been announced and accepted, but not learned.</p>
<pre>awplus# show ip bgp paths</pre>	<p>This command displays all BGP paths for IPv4 routes in the table.</p>
<pre>awplus# show bgp ipv6 paths</pre>	<p>This command displays all BGP4+ paths for IPv6 routes in the table.</p>
<pre>awplus# show ip bgp summary</pre>	<p>This command displays the status of all BGP connections.</p>
<pre>awplus# show bgp ipv6 summary</pre>	<p>This command displays the status of all BGP4+ connections.</p>

BGP Neighbor status logging

To enable the logging of messages generated when a BGP neighbor resets, use the following command in Router Configuration mode:

<pre>awplus(config-router)# bgp log-neighbor-changes</pre>	<p>This command generates log messages whenever a BGP neighbor resets.</p>
--	--

BGP Connections Reset

When a link between external neighbors drops, a BGP session does not reset immediately. To reset a BGP session when an interface goes down, use this command in Router Configuration mode

```
awplus(config-router)#  
bgp fast-external-failover
```

 This command resets BGP sessions automatically.

BGP Versions

By default, BGP sessions begin using BGP Version 4 and negotiate downward to earlier versions as needed. To prevent negotiation and force the BGP version used to communicate with a neighbor, use the command below in Router Configuration mode:

```
awplus(config-router)#  
neighbor <neighborid> version <version>
```

 This command specifies the BGP version to use when communicating with a neighbor.

BGP Confederations

One way to reduce the iBGP mesh is to divide an autonomous system into multiple sub-autonomous systems and group them into a single confederation. To the outside world, the confederation looks like a single autonomous system.

Each autonomous system is fully meshed within itself, and has few connections to other autonomous systems in the same confederation. Though peers in different autonomous systems have eBGP sessions, they exchange routing information as iBGP peers.

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number. To configure a BGP confederation identifier, use the following command in Router Configuration mode:

```
awplus(router-config)#  
bgp confederation identifier <1-4294967295>
```

 This command configures a BGP confederation.

In order to treat the neighbors from other autonomous systems within the confederation as special eBGP peers, use the following command in Router Configuration mode:

```
awplus(router-config)#  
bgp confederation peers <1-4294967295>
```

 This command specifies the autonomous systems that belong to the confederation.

BGP Route Reflectors

BGP requires that all iBGP speakers be fully meshed. Another way to reduce the iBGP mesh is to configure a BGP route reflector, instead of configuring a BGP confederation.

With route reflectors, all iBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an iBGP peer is configured to be a route reflector responsible for passing iBGP learned routes to a set of iBGP neighbors.

The internal peers of the route reflector are divided into two groups: client peers and all the other routers in the autonomous system (non-client peers). A route reflector reflects routes between these two groups.

The route reflector and its client peers form a cluster. The non-client peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with iBGP speakers outside their cluster.

When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and non-client peers.
- A route from a non-client peer is advertised to all clients.
- A route from a client is advertised to all clients and non-client peers. Hence, the clients need not be fully meshed.

To configure a route reflector and its clients, use the command below in Router Configuration mode:

```
awplus(router-config)#  
neighbor <neighborid> This command configures the local router as a BGP route  
route-reflector-client reflector and the specified neighbor as a client.
```

Along with route reflector-aware BGP speakers, you can have BGP speakers that do not understand the concept of route reflectors. They can be members of client or non-client groups allowing a gradual migration from the old BGP model to the route reflector model.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group or non-client group.

Each route reflector would be configured with other route reflectors as non-client peers, so the route reflectors will be fully meshed. The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

If the cluster has more than one route reflector, configure the cluster ID by using the following command in Router Configuration mode:

```
awplus(router-config)#  
bgp cluster-id {<ip-address> | This command configures the cluster-id if the BGP cluster has  
                           <cluster-id>} more than one route reflector. A cluster includes one or more  
                           route reflectors and their clients. Usually, each cluster is  
                           identified by the router-id of its single route reflector. .
```

Use the **show ip bgp (BGP)** command in Privileged Exec mode to display the originator ID and the cluster-list attributes.

BGP Community Filtering

BGP supports transit policies via controlled distribution of routing information. The distribution of routing information is based on one of the following three values:

- IP address.
- The value of the autonomous system path attribute.
- The value of the communities attribute (as described in this section).

The communities attribute is a way to group destinations into communities and apply routing decisions based on the communities. This method simplifies the configuration of a BGP speaker that controls distribution of routing information.

A community is a group of destinations that share some common attribute. Each destination can belong to multiple communities.

The communities attribute is an optional, transitive, global attribute in the numerical range from 1 to 4,294,967,200. Along with Internet community, there are a few predefined, well-known communities, as follows:

- **internet**—Advertise this route to the Internet community. All routers belong to it.
- **no-export**—Do not advertise this route to eBGP peers.
- **no-advertise**—Do not advertise this route to any peer (internal or external).
- **local-as**—Do not advertise this route to peers outside the local autonomous system. This route will not be advertised to other autonomous systems or sub-autonomous systems when confederations are configured.

Based on the community, you can control which routing information to accept, prefer, or distribute to other neighbors. When routes are aggregated, the resulting aggregate has a communities attribute that contains all communities from all the initial routes.

You can use community lists to create groups of communities to use in a match clause of a route map, and a series of community lists can be created. Statements are checked until a match is found. As soon as one statement is satisfied, the test is concluded.

To create a community list, use the following command in Global Configuration mode:

```
awplus(config)#
ip community-list <listname> {deny|permit} .<community>
```

This command creates a community-list. To set communities attributes and match clauses based on communities, see the [match community \(Route Map\)](#) command and the [set community \(Route Map\)](#) command for further information.

By default, the communities attribute is sent to a neighbor. To specify that the communities attribute are not sent to the neighbor, use the command below in Router Configuration mode:

```
awplus(config-router)#
no neighbor <neighborid>
send-community {both|extended|standard}
```

This command specifies that the communities attribute be sent to the neighbor at this IP address. Both standard and extended communities can be specified with the both keyword. Only standard or only extended can be specified with the standard and extended keywords

BGP Optimization

How to determine Backdoor Routes

You can determine which networks are reachable by using a backdoor route that the border router should use. A backdoor network is treated as a local network, except that it is not advertised. To configure backdoor routes, use the network backdoor command, beginning in Router Configuration mode:

```
awplus(config-router)#  
network {<ip-prefix/length> | This command determines reachable networks through  
    <ip-network-addr>} [mask backdoor routes when you apply the optional backdoor  
    <network-mask>] [route-map parameter.  
    <route-map-name>] [backdoor]
```

How to set the BGP Administrative Distance

Administrative Distance is a measure of the preference of different routing protocols. BGP uses three different administrative distances: external, internal, and local. Routes learned through external BGP are given the external distance, routes learned with iBGP are given the internal distance, and routes that are part of this autonomous system are given the local distance. To assign a BGP administrative distance, use the following command in Router Configuration mode:

```
awplus(config-router)#  
distance bgp <ebgp> <ibgp> This command assigns a BGP administrative distance.  
    <local>
```

Changing administrative distance of BGP routes may cause issues so is not recommended. The external distance should be lower than any other dynamic routing protocol, and the internal and local distances should be higher than any other dynamic routing protocol.

How to change BGP default local preference value

You can define a particular path as more preferable or less preferable than other paths by changing the default local preference value of 100. To assign a different default local preference value, use the following command in Router Configuration mode:

```
awplus(config-router)#  
bgp default local-preference This command changes the default local preference value.  
    <pref-value> You can use route maps to change the default local preference  
of specific paths.
```

How to redistribute the default network 0.0.0.0

By default, you are not allowed to redistribute network 0.0.0.0. To permit the redistribution of network 0.0.0.0, use the following command in Router Configuration or Address Family Configuration modes:

```
awplus(config-router)#
  neighbor {<neighborid>}
  default-originate [route-map
    <routemap-name>]
```

This command allows the redistribution of the default network 0.0.0.0 into BGP from the Router Configuration mode.

```
awplus(config-router-af)#
  neighbor {<neighborid>}
  default-originate [route-map
    <routemap-name>]
```

This command allows the redistribution of the default network 0.0.0.0 into BGP from the Address Family Configuration mode.

How to adjust BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which software declares a peer dead.

By default, the keepalive timer is 60 seconds, and the hold-time timer is 180 seconds. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated hold time and the configured keepalive time.

To adjust BGP timers for all neighbors, use the following command in Router Configuration mode:

```
awplus(config-router)#
  timers bgp <keepalive>
    <holdtime>
```

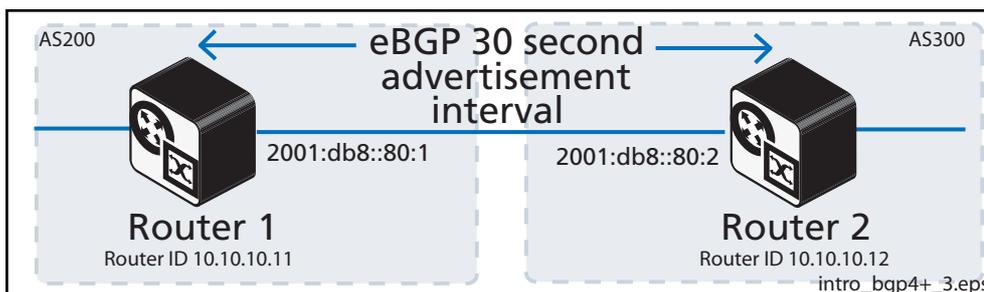
This command adjusts BGP timers for all neighbors.

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in Router Configuration mode:

```
awplus(config-router)#
  neighbor <neighborid> timers
  {<keepalive> <holdtime>|connect
    <connect>}
```

This command sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group. To clear the timers for a BGP neighbor or peer group, use the negated form of the neighbor timers command.

How to adjust the eBGP advertisement interval



When a route advertised by BGP changes, BGP sends an Update message. If an advertised route is flapping, and becomes unavailable. Route flapping is often caused when an interface is unstable, and a flood of Update messages will occur. One approach to control the flooding of BGP messages is to configure the advertisement interval. With the default interval value of 30 seconds for eBGP neighbors, BGP routing updates are sent only every 30 seconds, even if a route is flapping many times during this 30 second interval.

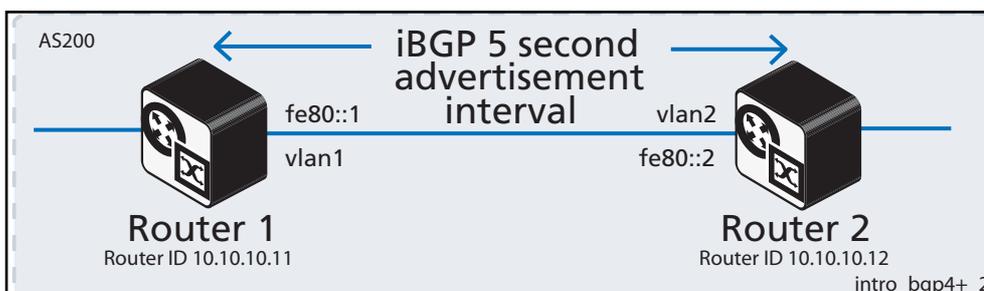
To assign a different **advertisement-interval** value for eBGP peers, use the following command in Router Configuration mode::

```
awplus(config-router)#
neighbor <neighborid> advertisement-interval <time>
```

This command sets the minimum interval between sending eBGP routing updates with a 30 second eBGP default interval.

Use this command instead of the **neighbor as-origination-interval (BGP and BGP4+)** command for eBGP peers with prefixes not in the same AS for updates not in a local AS.

How to adjust the iBGP advertisement interval



With the default value of 5 seconds for iBGP neighbors, BGP routing updates are sent only every 5 seconds, even if a route is flapping many times during this 5 second interval.

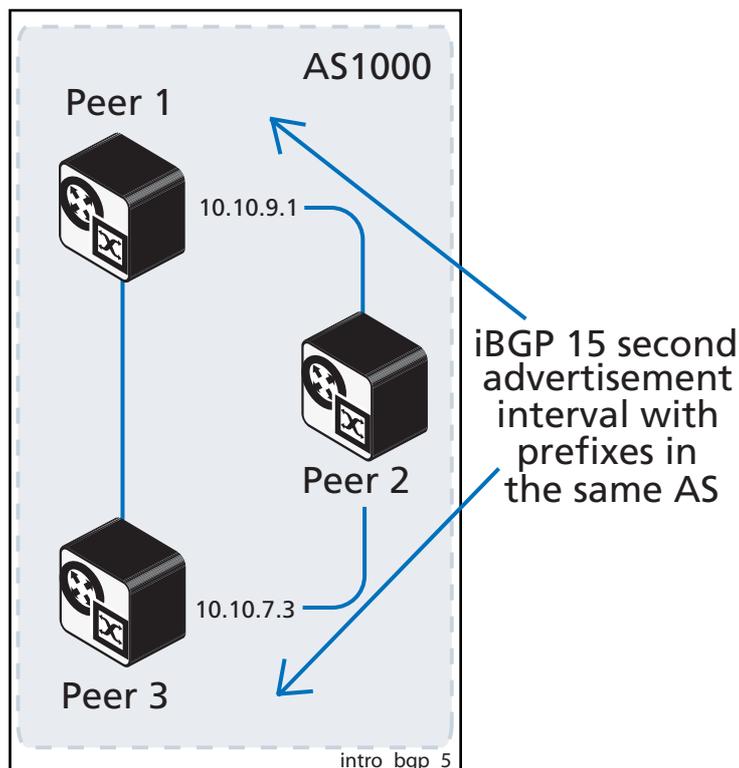
To assign a different **advertisement-interval** value for iBGP peers, use the following command in Router Configuration mode :

```
awplus(config-router)#
neighbor <neighborid> advertisement-interval <time>
```

This command sets the minimum interval between sending eBGP routing updates with a 5 second iBGP default interval.

BGP dampening can also be used to control the effects of flapping routes. See the **BGP and BGP4+ Route Dampening** section in this chapter, and the **bgp dampening (BGP and BGP4+)** command in the **BGP and BGP4+ Commands** chapter for more information.

How to adjust the AS origination interval



With the default value of 15 seconds for iBGP neighbors with a prefix in the same AS, BGP routing updates are sent only every 15 seconds, even if a route is flapping many times during this 15 second interval.

To assign a different **as-origination-interval** value for iBGP peers with a prefix in the same AS, use the following command in Router Configuration mode: :

```
awplus(config-router)#
neighbor <neighbor_address> as-origination-interval <time>
```

This command sets the minimum interval between sending iBGP routing updates with a 15 second iBGP default interval to report changes in the local AS for iBGP peers with prefixes in the same local AS.

Use this command instead of the **neighbor advertisement-interval (BGP and BGP4+)** command for iBGP peers with prefixes in the same AS for updates only within a local AS.

BGP dampening can also be used to control the effects of flapping routes. See the **BGP and BGP4+ Route Dampening** section in this chapter, and the **bgp dampening (BGP and BGP4+)** command in the **BGP and BGP4+ Commands** chapter for more information.

How to remove private AS numbers for BGP

There are public and private AS numbers, in the same way there are public and private IP addresses. Like the private IP addresses that you cannot send to the Internet, you cannot send private AS numbers to the Internet. This functionality enables you to remove them.

Public AS numbers range from 1 to 64,511. Private AS numbers range for 64,512 and 65,535. Private AS numbers are used to divide large ASs into smaller multiple ASs for eBGP.

ISPs may assign private AS numbers to save public AS number usage. When you receive private AS numbers, you need to remove them before you send updates to the Internet. To remove private AS numbers for BGP, use this command in the Router Configuration mode:

```
awplus(config-router)#  
neighbor <neighborid> remove-private-AS
```

This command removes private AS numbers for BGP in Router Configuration mode. This command is not supported for BGP4+ in IPv6 Address Family Configuration mode.

This command removes a private AS number and makes an update packet with a public AS number as the AS path attribute. So only public AS numbers are entered in Internet BGP routing tables, and private AS numbers are not entered in Internet BGP tables.

For the filtering to apply, both peering devices must be set to use either 2-byte or extended 4- byte ASN (with the same ASN type set on both peers). For example, if a device (which defaults to use a 4-byte ASN), is peered with a device that defaults to a 2-byte ASN, then the device using a 2-byte ASN device also needs to be configured with the command **bgp extended-asn-cap** for the filtering to apply.

The **neighbor remove-private-AS (BGP)** command is not available in IPv6 Address Family Configuration mode for BGP4+. This command is only available in Router Configuration mode for BGP.

Verify the application of this command with the **show running-config** and **show ip bgp** commands. See sample verification output in the following sections:

- [How to remove BGP filtering for private AS numbers](#)
- [How to apply BGP filtering for private AS numbers.](#)

How to remove BGP filtering for private AS numbers

Remove BGP filtering for private AS numbers using the `no neighbor <neighborid> remove-private-AS` command. See this example from Router Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 5
awplus(config-router)# no neighbor 192.168.0.2 remove-private-AS
```

If the AS path attribute command is removed from the Router 2 BGP configuration, then the private AS is correctly not filtered. See the sample console command entries below:

```
Router 2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router 2(config)#router bgp 5
Router 2(config-router)#no neighbor 192.168.0.2 remove-private-AS
Router 2(config-router)#exit
Router 2(config)#exit
Router 2#clear ip bgp *
```

The `show ip bgp` command output below on Router 1 shows an update for the network 10.0.0.0/24 received from Router 2 (192.68.0.1) has an AS path attribute 5, which is the AS number of Router 2, and contains private AS path 65000.

```
Router 1#show running-config router bgp
!
router bgp 1
neighbor 192.168.0.1 remote-as 5
!
Router 1#show ip bgp
BGP table version is 1017, local router ID is 192.168.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
              l - labeled, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 10.0.0.0/24      192.168.0.1         0           0 5 65000 i
*> 172.30.0.0       192.168.0.1         0           0 5 i
*> 192.168.0.0     192.168.0.1         0           0 5 i
```

How to apply BGP filtering for private AS numbers

Apply BGP filtering for private AS numbers using the `neighbor <neighborid> remove-private-AS` command. See this example from Router Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 5
awplus(config-router)# neighbor 192.168.0.2 remove-private-AS
```

If the non-default AS path attribute command is applied to the Router 2 BGP configuration then the private AS is correctly filtered. See the sample console command entries below:

```
Router 2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router 2(config)#router bgp 5
Router 2(config-router)#neighbor 192.168.0.2 remove-private-AS
Router 2(config-router)#exit
Router 2(config)#exit
Router 2#clear ip bgp *
```

The `show ip bgp` command output on ISP Router 1 shows that the update for the network 10.0.0.0/24 received from Router 2 (192.68.0.1) has an AS path attribute 5. This is the AS number of Router 2, without private AS path 65000.

In the ISP router BGP prefix table below, the only AS path attribute for the 10.0.0.0/24 network is '5', and the AS 65000 has been correctly filtered out from the update sent by Router 2 received into ISP Router 1. This is displayed in the `show ip bgp` output below:

```
Router 1#show running-config router bgp
!
router bgp 1
neighbor 192.168.0.1 remote-as 5
neighbor 192.168.0.2 remove-private-as
!
Router 1#show ip bgp
BGP table version is 1041, local router ID is 192.168.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
              l - labeled, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 10.0.0.0/24      192.168.0.1         0         0 5 i
*> 172.30.0.0       192.168.0.1         0         0 5 i
*> 192.168.0.0      192.168.0.1         0         0 5 i
```

BGP and BGP4+ Terminology

See the below table for descriptions of BGP and BGP4+ terms used through this chapter:

Term	Description
AS number (ASN)	A number that identifies an Autonomous System. Previously, an ASN was a 16-bit number between 1 and 64,511 (public) and 64,512 and 65,535 (private) assigned to an AS for the purpose of BGP operation. Since RFC 6793, an ASN is now 32-bit with a range of decimal values between 1 and 4,294,967,295 .
AS_Path	A BGP path attribute that lists ASNs through which the route has been advertised.
AS_Sequence	A type of AS_Path segment consisting of an ordered list of ASNs through which the route has been advertised.
auto summary	A routing protocol feature in which a router that connects to more than one classful network advertises summarized routes for each entire classful network when sending updates out interfaces connected to other classful networks.
Autonomous System (AS)	A set of routers inside a single administrative authority, grouped together for the purpose of controlling routing policies for the routes advertised by that group to the Internet.
BGP best path algorithm	The process that BGP uses to examine competing BGP paths (routes) in its BGP table, for a single prefix, choosing one route as the best route. The best path algorithm results in a choice of a single route for each prefix as that router's best BGP path. BGP compares the current best path against the next path in the list.
BGP peer	A BGP neighbor. A BGP neighbor is a router running BGP that formed a BGP neighbor relationship to exchange BGP Updates.
BGP peer group	A configuration construct in which multiple neighbors' parameters can be configured as a group to reduce the length of the configuration. BGP performs routing policy logic against one set of Updates for the peer group, improving convergence time and reducing the CPU load.
BGP synchronization	A feature in which BGP routes cannot be considered to be a best route to reach a prefix unless that same prefix exists in the router's IP routing table as learned via some IGP. When synchronization is enabled, BGP does not advertise a route learned via an iBGP peer to external peers, until the route is learned locally, or via an IGP and is present in the routing table. When synchronization is disabled, BGP can use and advertise routes from an iBGP peer to external peers, which are not present in the routing table.
BGP table	A table inside a router that holds the path attributes known by BGP on that router.
Border Gateway Protocol (BGP)	An exterior routing protocol designed to exchange prefix information between different autonomous systems. The information includes characteristics called path attributes, which allow for great flexibility regarding routing choices.

Term	Description
CIDR	Classless Inter-Domain Routing. A routing method supported by BGP that is based on route aggregation. CIDR allows router to group routes together to cut down on the amount of routing information carried by routers. Several IP networks appear as a single network to networks outside of those grouped together.
External BGP (eBGP)	Refers to how a router views a BGP peer relationship, where the peer is in another AS (Autonomous System). See also Internal BGP (iBGP), where the peer is in the same AS.
Interior Gateway Protocol (IGP)	A routing protocol designed to be used to exchange routing information inside a single autonomous system.
Internal BGP (iBGP)	A characteristic of a BGP neighbor relationship, specifically when the two routers are internal to the same BGP ASN.
neighbor	Another router with which a router decides to exchange routing information.
Neighbor Type	Either external BGP (eBGP), confederation eBGP, or internal BGP (iBGP). The term refers to a peer connection and whether the peers are in different ASs (eBGP), different confederation systems (confederation eBGP), or in the same AS (iBGP).
Next Hop field	With a routing update, or routing table entry, the portion of a route that defines the next router to which a packet should be sent to reach the destination subnet. With routing protocols, the Next Hop field may define a router other than the router sending the routing update.
Next-hop self	A BGP configuration setting that tells the local router to change the NEXT_HOP path attribute to refer to its own BGP Update Source when advertising routes to BGP neighbors.
path control	A general term, that refers to any function that impacts how routers forward packets. These functions include routing protocols and any other feature that impacts the IP routing table, plus any feature that impacts the packet forwarding process.
private AS	BGP ASN whose value is between 64,512 and 65,535. These values are not assigned for use on the Internet and can be used for private purposes, typically either within confederations or by ISPs to hide the ASN used by some customers.
A private ASN	An Autonomous System Number (ASN) that falls inside the Private AS range.
public ASN	An ASN that fits below the private ASN range, specifically from 1 through 54,511 with a 16-bit ASN. With a 32-bit ASN the range is 1 through 4,294,967,295.
redistribution	The process on a router of taking the routes from the IP routing table, as learned by one routing protocol, and injecting routes for those same subnets into another routing protocol.
regular expression	A list of interspersed alphanumeric literals and metacharacters used to apply complex matching logic to alphanumeric strings. Often used for matching AS_Paths.

Term	Description
route map	Enables logic to be applied to a set of items. Often used for decisions about what routes to redistribute and for setting characteristics of those routes, for instance, metric values.
route redistribution	The process of taking routes known through one routing protocol and advertising those routes with another routing protocol.
route summarization	A consolidation of advertised addresses that causes a single summary route to be advertised.
summary route	A route that is created to represent one or more smaller component routes, typically to reduce the size of routing and topology tables.

Chapter 2: BGP and BGP4+ Configuration



Introduction	2.2
BGP Configurations	2.3
Enabling BGP Peers In The Same Autonomous System	2.3
Enabling BGP Between Different Autonomous Systems	2.5
Configuring a BGP Route Reflector	2.6
Configuring a BGP Confederation	2.8
Configuring BGP Authentication	2.11
Configuring BGP Graceful Reset	2.13
Configuring BGP Distance	2.15
Configuring BGP Graceful Restart	2.18
Configuring BGP Weight per Peer	2.20
BGP4+ Configurations	2.23
Enabling iBGP Peering using a Global Address	2.23
Enabling iBGP Peering using a Link-local Address	2.26
Enabling eBGP Peering between different Autonomous Systems	2.29
Configuring Route-Maps with BGP4+	2.31
Configuring BGP4+ Graceful Restart	2.35
Configuring BGP4+ Distance	2.39
Configuring BGP4+ Graceful Reset	2.43

Introduction

This chapter contains basic BGP and BGP4+ configuration examples.

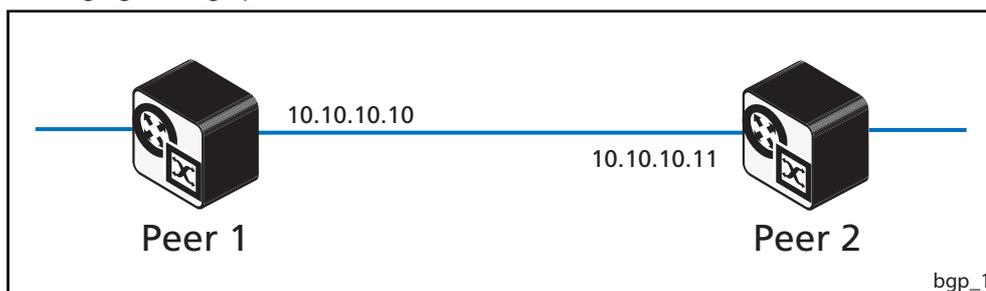
For details about the commands used in these examples, or the outputs from validation commands, see [Chapter 3, BGP and BGP4+ Commands](#). This chapter provides an alphabetical reference of commands used to configure the Border Gateway Protocol for IPv4 (BGP) and for IPv6 (BGP4+).

For basic BGP and BGP4+ introduction information, including terminology and concepts about BGP and BGP4+, including the concept of autonomous system numbers (ASN), path attributes (PA), and both internal and external BGP/BGP4+, see [Chapter 1, BGP and BGP4+ Introduction](#).

BGP Configurations

Enabling BGP Peers In The Same Autonomous System

This example shows the minimum configuration required for enabling BGP on an interface. **Peer 1** and **Peer 2** are two BGP Peers belonging to the same Autonomous System (AS), AS200, connecting to network 10.10.10.0/24. First, define the routing process and the AS number to which the peers belong. Then, define BGP neighbors to start exchanging routing updates.



Peer 1

```
awplus#
configure terminal  Enter the Global Configuration mode.

awplus(config)#
router bgp 200      Define the BGP routing process. The number 200 specifies
                   the AS number of Peer 1.

awplus(config-router)#
neighbor 10.10.10.11 remote-as 200  Define BGP neighbors. 10.10.10.11 is the IP address of the
                                     neighbor (Peer 2) and 200 is the neighbor's AS number.
```

Peer 2

```
awplus#
configure terminal  Enter the Global Configuration mode

awplus(config)#
router bgp 200      Define the BGP routing process. The number 200 specifies
                   the AS number of Peer 2.

awplus(config-router)#
neighbor 10.10.10.10 remote-as 200  Define BGP neighbors. 10.10.10.10 is the IP address of the
                                     neighbor (Peer 1) and 200 is the neighbor's AS number.
```

Names of Commands Used

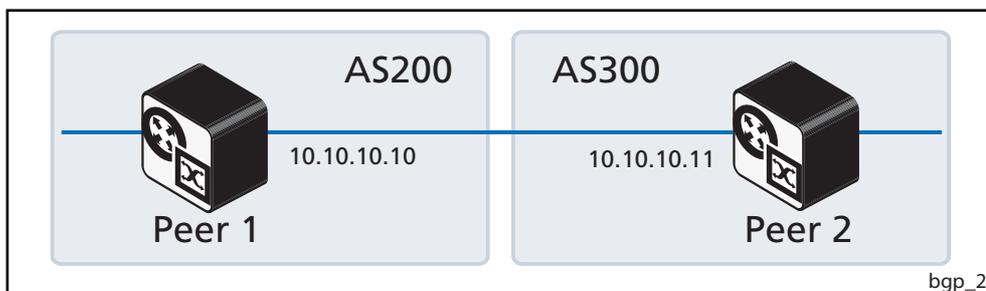
router bgp (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)

Validation Commands

show ip bgp summary (BGP)
show ip bgp neighbors (BGP)

Enabling BGP Between Different Autonomous Systems

This example shows the minimum configuration required for enabling BGP on an interface when the BGP peers belong to different Autonomous Systems (ASs). **Peer 1** and **Peer 2** are two BGP peers in different ASs, AS200 and AS300 connecting to network 10.10.10.0/24.



Peer 1

```
awplus#
configure terminal Enter the Global Configuration mode.

awplus(config)#
router bgp 200 Define the BGP routing process. The number 200
specifies the AS number of Peer 1.

awplus(config-router)#
neighbor 10.10.10.11 remote-as 300 Define BGP neighbors. 10.10.10.11 is the IP address of the
neighbor (Peer 2) and 300 is the neighbor's AS number.
```

Peer 2

```
awplus#
configure terminal Enter the Global Configuration mode.

awplus(config)#
router bgp 300 Define the BGP routing process. The number 300
specifies the AS number of Peer 2.

awplus(config-router)#
neighbor 10.10.10.10 remote-as 200 Define BGP neighbors. 10.10.10.10 is the IP address of the
neighbor (Peer 1) and 200 is the neighbor's AS number.
```

Names of Commands Used

router bgp (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)

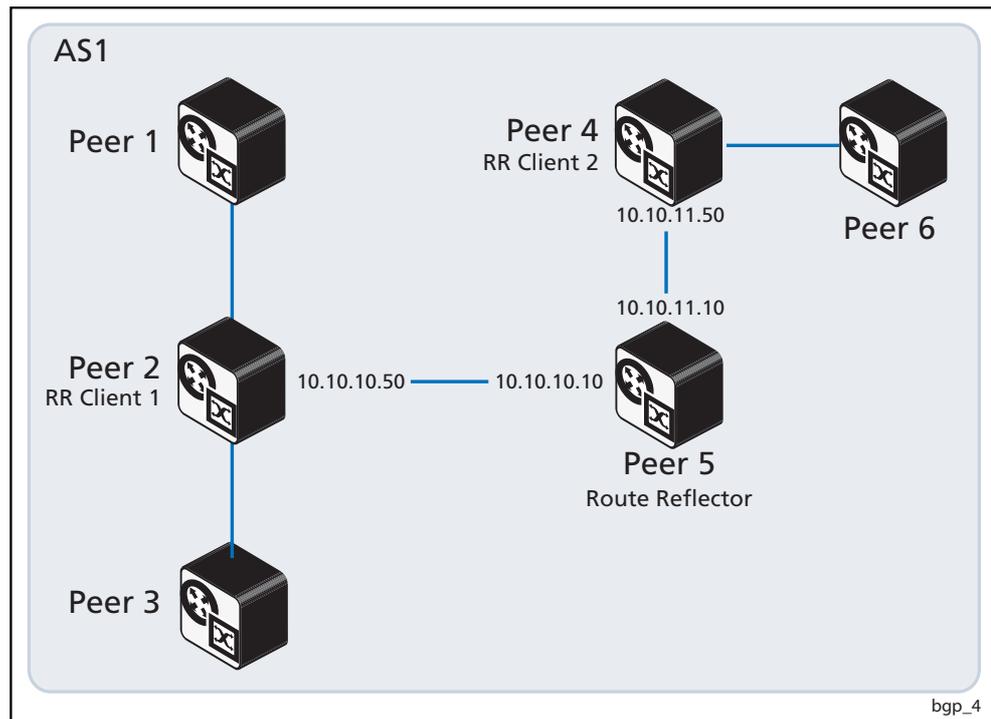
Validation Commands

show ip bgp summary (BGP)
show ip bgp neighbors (BGP)

Configuring a BGP Route Reflector

Use a BGP Route Reflector to reduce the iBGP mesh inside an AS. In this example, Peer 2, Peer 5 and Peer 4 would have to maintain a full mesh among themselves but by making Peer 5 the Route Reflector, Peer 2 (Client1) has an iBGP session with the Route Reflector only and not with Peer 4 (Client 2).

The routes learned from Peer 2 are advertised to the other clients and to iBGP peers outside the cluster; the iBGP routes learned from iBGP peers outside the cluster are advertised to Peer 2. This reduces the iBGP peer connections in AS1.



Route Reflector (Peer 5)

<code>awplus#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config)#</code>	
<code>router bgp 1</code>	Define the BGP routing process. The number 1 specifies the AS number of Peer 5 .
<code>awplus(config-router)#</code>	
<code>neighbor 10.10.10.50 remote-as 1</code>	Define BGP neighbors. 10.10.10.50 is the IP address of one of the neighbors (Peer 2) and 1 is the neighbor's AS number.
<code>awplus(config-router)#</code>	
<code>neighbor 10.10.10.50 route-reflector client</code>	Configure Peer 5 as the Route-Reflector (RR) and neighbor Peer 2 as its client.
<code>awplus(config-router)#</code>	
<code>neighbor 10.10.11.50 remote-as 1</code>	Define BGP neighbor. 10.10.11.50 is the IP address of one of the neighbors (Peer 4) and 1 is the neighbor's AS number.

```
awplus(config-router)#
neighbor 10.10.11.50 route-reflector
                        client
```

Configure **Peer 5** as the Route-Reflector (RR) and neighbor **Peer 4** as its client.

Route Reflector Client 1 (Peer 2)

```
awplus(config)#
router bgp 1
```

Define the BGP routing process. The number 1 specifies the AS number of **Peer 2**.

```
awplus(config-router)#
neighbor 10.10.10.10 remote-as 1
```

Define BGP neighbors. 10.10.10.10 is the IP address of the neighbor (**Peer 5**) and 1 is the neighbor's AS number.

Route Reflector Client 2 (Peer 4)

```
awplus(config)#
router bgp 1
```

Define the BGP routing process. The number 1 specifies the AS number of **Peer 4**.

```
awplus(config-router)#
neighbor 10.10.11.10 remote-as 1
```

Define BGP neighbors. 10.10.11.10 is the IP address of the neighbor (**Peer 5**) and 1 is the neighbor's AS number.

Names of Commands Used

neighbor remote-as (BGP and BGP4+)
neighbor route-reflector-client (BGP)

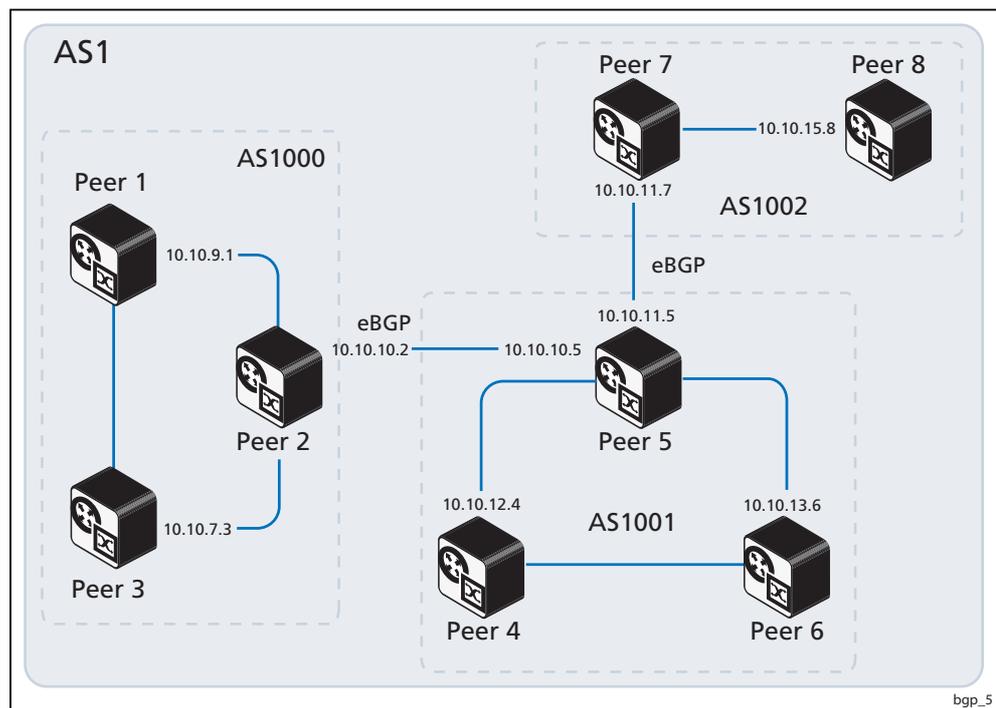
Validation Commands

show ip bgp (BGP)
show ip bgp neighbors (BGP)

Configuring a BGP Confederation

In this example, AS1 contains three Confederation Autonomous Systems - AS 1000, AS 1001 and AS 1002. To any outside AS, the Confederation is a single AS. AS1 Confederation eBGP is run between Peer 2 and Peer 5, and between Peer 5 and Peer 7. Peer 2 is configured so that its local AS is 1000. Its peer connection to Peer 5 is set up like any other eBGP session.

The **bgp confederation identifier (BGP and BGP4+)** command on page 3.21 tells the peer that it is a member of a Confederation and the Confederation ID. The **bgp confederation peers (BGP and BGP4+)** command on page 3.22 lists the member AS to which Peer 2 is connected. The command tells the BGP process that the eBGP connection is a Confederation eBGP rather than a normal eBGP.



Peer 2

<code>awplus#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config)#</code>	
<code>router bgp 1000</code>	Define the BGP routing process. The number 1000 specifies the AS number of Peer 2 .
<code>awplus(config-router)#</code>	
<code>bgp confederation identifier 1</code>	Specify a BGP Confederation Identifier. To routers outside the confederation, the whole confederation will appear as a single AS, and the identifier will appear as its AS number.
<code>awplus(config-router)#</code>	
<code>bgp confederation peers 1001 1002</code>	Specify 1001 (AS1001) and 1002 (AS1002) to become members of the Confederation.

Peer 2 (cont'd)

```

awplus(config-router)#
neighbor 10.10.10.5 remote-as 1001
awplus(config-router)#
neighbor 10.10.9.1 remote-as 1000
awplus(config-router)#
neighbor 10.10.7.3 remote-as 1000

```

Define BGP neighbors for **Peer 2** by specifying the IP addresses and the AS numbers of neighbors.

Peer 5

```

awplus(config)#
router bgp 1001

```

Define the BGP routing process. The number 1001 specifies the AS number of **Peer 5**.

```

awplus(config-router)#
bgp confederation identifier 1

```

Specify a BGP Confederation Identifier. To routers outside the confederation, the whole confederation will appear as a single AS, and the identifier will appear as its AS number.

```

awplus(config-router)#
bgp confederation peers 1000 1002

```

Specify 1000 (**AS1000**) and 1002 (**AS1002**) to become members of the Confederation.

```

awplus(config-router)#
neighbor 10.10.10.2 remote-as 1000
awplus(config-router)#
neighbor 10.10.11.7 remote-as 1002
awplus(config-router)#
neighbor 10.10.13.6 remote-as 1001
awplus(config-router)#
neighbor 10.10.12.4 remote-as 1001

```

Define BGP neighbors for **Peer 5** by specifying the IP addresses and the AS numbers of neighbors.

Peer 7

<code>awplus(config)#</code>	
<code>router bgp 1002</code>	Define the BGP routing process. The number 1001 specifies the AS number of Peer 5 .

<code>awplus(config-router)#</code>	
<code> bgp confederation identifier 1</code>	Specify a BGP Confederation Identifier. To routers outside the confederation, the whole confederation will appear as a single AS, and the identifier will appear as its AS number.

<code>awplus(config-router)#</code>	
<code> bgp confederation peers 1000 1001</code>	Specify 1000 (AS1000) and 1001 (AS1001) to become members of the Confederation.

<code>awplus(config-router)#</code>	
<code>neighbor 10.10.11.5 remote-as 1001</code>	Define BGP neighbors for Peer 7 by specifying the IP addresses and the AS numbers of neighbors.
<code> awplus(config-router)#</code>	
<code>neighbor 10.10.15.8 remote-as 1002</code>	

Names of Commands Used

neighbor remote-as (BGP and BGP4+)
bgp confederation peers (BGP and BGP4+)
bgp confederation identifier (BGP and BGP4+)

Validation Commands

show ip bgp (BGP)
show ip bgp neighbors (BGP)

Configuring BGP Authentication

BGP authentication allows users to receive selected routing information, enhancing security of their network traffic. When BGP authentication is enabled on a peer, the peer verifies routing packet it receives by exchanging a password that is configured on both the sending and the receiving peers.

In this example, both **Peer 1** and **Peer 2** have `manager` as the password. Configure the same password on all routers that are to communicate using BGP in a network.



Peer 1

```
awplus(config)#
configure terminal Enter the Global Configuration mode.

awplus(config-router)#
router bgp 200 Define the BGP routing process. The number 200
specifies the AS number of Peer 1.

awplus(config-router)#
neighbor 10.10.10.11 remote-as 200 Define BGP neighbors and establish a TCP session.
10.10.10.11 is the IP address of the neighbor (Peer 2)
and 200 is the neighbor's AS number.

awplus(config-router)#
neighbor 10.10.10.11 password 1 Specify the encryption type and the password.
manager
```

Peer 2

```
awplus(config)#
configure terminal Enter the Global Configuration mode.

awplus(config-router)#
router bgp 200 Define the BGP routing process.
The number 200 specifies the AS number of Peer 2.

awplus(config-router)#
neighbor 10.10.10.10 remote-as 200 Define BGP neighbors and establish a TCP session.
10.10.10.10 is the IP address of the neighbor (Peer 1)
and 200 is the neighbor's AS number.

awplus(config-router)#
neighbor 10.10.10.10 password 1 Specify the encryption type and the password.
manager
```

Names of Commands Used

neighbor remote-as (BGP and BGP4+)
neighbor password (BGP and BGP4+)

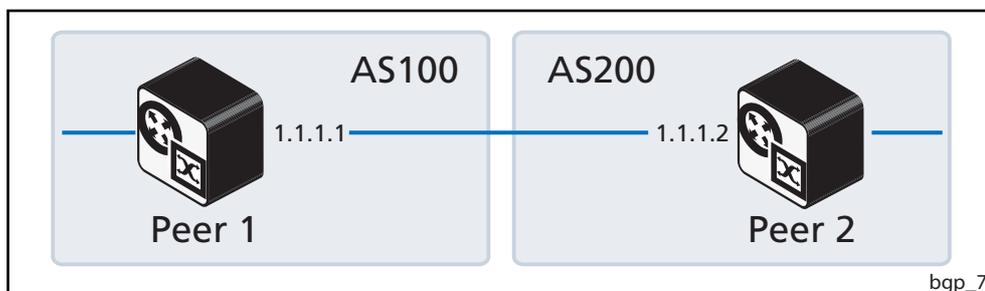
Validation Commands

show ip bgp (BGP)
show ip bgp neighbors (BGP)

Configuring BGP Graceful Reset

The graceful restart feature for BGP session reset is used so that any changes in network configuration do not affect packet forwarding. The **bgp graceful-restart graceful-reset (BGP and BGP4+)** command on page 3.33 invokes graceful restart only when a configuration change forces a peer reset.

This example shows configuring BGP graceful reset.



Peer 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP routing process. The number 100 specifies the AS number of Peer 1.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart</code>	Enable BGP graceful restart support.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart graceful-reset</code>	Configure to invoke graceful restart when a configuration change forces a peer reset.
<code>awplus(config-router)#</code>	
<code>neighbor 1.1.1.2 remote-as 200</code>	Specify the neighbor's IP address (1.1.1.2) and the AS number of the neighbor (200).
<code>awplus(config-router)#</code>	
<code>neighbor 1.1.1.2 capability graceful-restart</code>	Specify the neighbor's IP address (1.1.1.2) for which the graceful restart capability is supported.

Peer 2

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 200</code>	Define the BGP routing process. The number 200 specifies the AS number of Peer 2 .
<code>awplus(config-router)# bgp graceful-restart</code>	Enable BGP graceful restart support.
<code>awplus(config-router)# bgp graceful-restart graceful-reset</code>	Configure to invoke graceful restart when a configuration change forces a peer reset.
<code>awplus(config-router)# neighbor 1.1.1.1 remote-as 100</code>	Specify the neighbor's IP address (1.1.1.1) and the AS number of the neighbor (100).
<code>awplus(config-router)# neighbor 1.1.1.1 capability graceful-restart</code>	Specify the neighbor's IP address (1.1.1.1) for which the graceful restart capability is supported

Names of Commands Used

bgp graceful-restart (BGP and BGP4+)
bgp graceful-restart graceful-reset (BGP and BGP4+)
neighbor capability graceful-restart (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
neighbor restart-time (BGP and BGP4+)
router bgp (BGP and BGP4+)

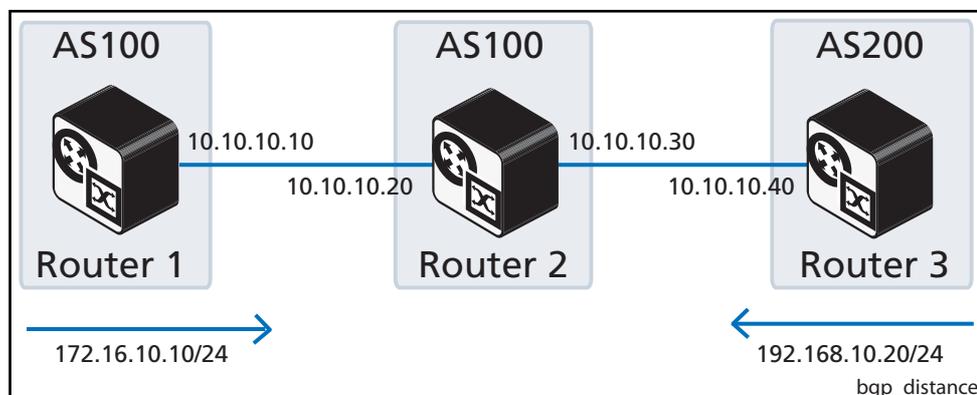
Validation Commands

show ip bgp (BGP)
show ip bgp neighbors (BGP)
show ip bgp summary (BGP)

Configuring BGP Distance

You can configure administrative distance in BGP for a specific address family.

This example shows configuring the BGP distance for the IPv4 Address Family.



Router 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP routing process. The number 100 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>network 172.16.10.10/24</code>	Specify the network 172.16.10.10/24 to be advertised by the BGP routing process from Router 1 to Router 2 .
<code>awplus(config-router)#</code>	
<code>neighbor 10.10.10.20 remote-as 100</code>	Specify the neighbor's IPv4 address (10.10.10.20) and Autonomous System (AS) number (100). Router 2 is the neighbor of Router 1 .
<code>awplus(config-router)#</code>	
<code>address-family ipv4 unicast</code>	Enter the IPv4 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)#</code>	
<code>neighbor 10.10.10.20 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.20 for Router 2 . Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)#</code>	
<code>exit-address-family</code>	Exit IPv4 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)#</code>	
<code>exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 2

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 100</code>	Define the BGP routing process. The number 100 specifies the AS number of Router 2 .
<code>awplus(config-router)# neighbor 10.10.10.40 remote-as 200</code>	Specify the neighbor's IPv4 address (10.10.10.40) and Autonomous System (AS) number (200). Router 3 is a neighbor of Router 2 .
<code>awplus(config-router)# neighbor 10.10.10.10 remote-as 100</code>	Specify the neighbor's IPv4 address (10.10.10.10) (and Autonomous System (AS) number (100)). Router 1 is a neighbor of Router 2 .
<code>awplus(config-router)# address-family ipv4 unicast</code>	Enter the IPv4 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)# aggregate-address 172.31.10.10/24 summary-only</code>	Configure an IPv4 non-AS-set aggregate route on Router 2 . Note that the local distance gets applied to this route.
<code>awplus(config-router-af)# neighbor 10.10.10.10 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.10 (Router 1). Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)# neighbor 10.10.10.40 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.40 (Router 3). Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)# distance 12 13 11</code>	Configure the administrative distance for external, internal, and local routes received in IPv6 Address Family Configuration mode.
<code>awplus(config-router-af)# exit-address-family</code>	Exit IPv4 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)# exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 3

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 200</code>	Define the BGP routing process. The number 200 specifies the AS number of Router 3 .
<code>awplus(config-router)# network 192.168.10.20/24</code>	Specify the network 192.168.10.20/24 to be advertised by the BGP4+ routing process from Router 3 to Router 2 .
<code>awplus(config-router)# neighbor 10.10.10.30 remote-as 100</code>	Specify the neighbor's IPv4 address (10.10.10.30) (and Autonomous System (AS) number (100)). Router 2 is a neighbor of Router 3 .
<code>awplus(config-router)# address-family ipv4 unicast</code>	Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)# neighbor 10.10.10.30 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.30 for Router 2 . Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)# exit-address-family</code>	Exit IPv4 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)# exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

[address-family \(BGP and BGP4+\)](#)
[aggregate-address \(BGP and BGP4+\)](#)
[exit-address-family \(BGP and BGP4+\)](#)
[neighbor remote-as \(BGP and BGP4+\)](#)
[network \(BGP and BGP4+\)](#)
[router bgp \(BGP and BGP4+\)](#)

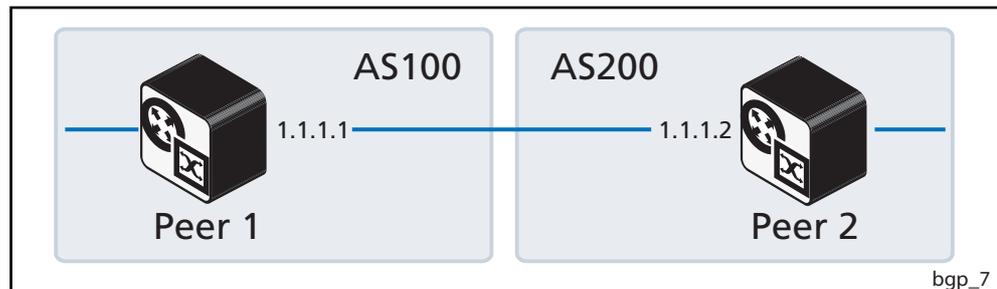
Validation Commands (re-link to BGP commands)

[show ip bgp \(BGP\)](#)
[show ip bgp neighbors \(BGP\)](#)
[show ip bgp summary \(BGP\)](#)

Configuring BGP Graceful Restart

Using BGP graceful restart, the data forwarding plane of the device can continue to process and forward packets even if the control plane, which is responsible for determining best paths, fails. The **bgp graceful-restart (BGP and BGP4+)** command on [page 3.32](#) sets the maximum time that a graceful-restart neighbor waits to come back up after a restart. This value is applied to all neighbors unless you explicitly override it by configuring the corresponding value on the neighbor.

This example shows configuring BGP graceful restart.



Peer 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP routing process. The number 100 specifies the AS number of Peer 1.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart restart-time 180</code>	Configure the maximum time in seconds (180) required for neighbor(s) to restart.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart stalepath-time 150</code>	Configure the maximum time in seconds (150) to retain stale paths from the restarting neighbor(s).
<code>awplus(config-router)#</code>	
<code>bgp update-delay 120</code>	Configure the maximum time in seconds (120) to defer initial route selection.
<code>awplus(config-router)#</code>	
<code>neighbor 1.1.1.2 remote-as 200</code>	Specify the neighbor's IP address (1.1.1.2) and the AS number of the neighbor (200).
<code>awplus(config-router)#</code>	
<code>neighbor 1.1.1.2 capability graceful-restart</code>	Specify the neighbor's IP address (1.1.1.2) for which the graceful restart capability is supported.

Peer 2

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 200</code>	Define the routing process. The number 200 specifies the AS number of Peer 2 .
<code>awplus(config-router)# bgp graceful-restart</code>	Enable BGP graceful restart support. If the restart and stale-path timers are not configured, the defaults are 90 seconds and 360 seconds, respectively.
<code>awplus(config-router)# neighbor 1.1.1.1 remote-as 100</code>	Specify the neighbor's IP address (1.1.1.1) and the AS number of the neighbor (100).
<code>awplus(config-router)# neighbor 1.1.1.1 capability graceful-restart</code>	Specify the neighbor's IP address (1.1.1.1) for which the graceful restart capability is supported
<code>awplus(config-router)# neighbor 1.1.1.1 restart-time 160</code>	Configure the maximum time in seconds (160) required for the neighbor to restart. This overwrites the configured restart time for that neighbor. For example: the above configured restart-time for all neighbors is 90 seconds (the default). With this command, the neighbor, 1.1.1.1, will have a 160 second restart-time.

Names of Commands Used

bgp graceful-restart (BGP and BGP4+)
bgp graceful-restart graceful-reset (BGP and BGP4+)
bgp update-delay (BGP and BGP4+)
neighbor capability graceful-restart (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
neighbor restart-time (BGP and BGP4+)
router bgp (BGP and BGP4+)

Validation Commands

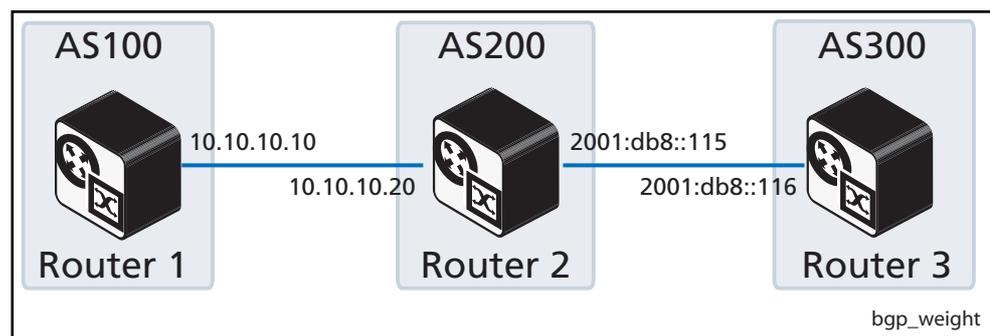
show ip bgp (BGP)
show ip bgp neighbors (BGP)
show ip bgp summary (BGP)

Configuring BGP Weight per Peer

You can assign a different weight per address family of a peer. For example, a network can be configured to prefer IPv6 routes from one peer and IPv4 routes from another peer.

If the neighbor weight command is given under a specific address family mode, then the peer weight is set for that specific address family. If the address family is not specifically set first, then the weight is updated for the default address family instead.

This example shows configuring the BGP Weight per Peer both for IPv4 Address Family and for IPv6 Address Family. Router 1 (AS100) uses IPv4 Address Family, Router 2 (AS200) uses both IPv4 and IPv6 Address Family, Router 3 (AS300) uses IPv6 Address Family.



Router 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP routing process. The number 100 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>address-family ipv4 unicast</code>	Enter the IPv4 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)#</code>	
<code>neighbor 10.10.10.20 remote-as 200</code>	Specify the neighbor's IPv4 address (10.10.10.20) and Autonomous System (AS) number (200). Router 2 is the neighbor of Router 1 .
<code>awplus(config-router-af)#</code>	
<code>neighbor 10.10.10.20 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.20 for Router 2 in IPv4 Address Family Configuration mode. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)#</code>	
<code>exit-address-family</code>	Exit IPv4 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)#</code>	
<code>exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 21

awplus(config)# configure terminal	Enter the Global Configuration mode.
awplus(config-router)# router bgp 200	Define the BGP routing process. The number 200 specifies the AS number of Router 2 .
awplus(config-router)# address-family ipv4 unicast	Enter the IPv4 Address Family Configuration mode from the Global Configuration mode.
awplus(config-router-af)# neighbor 10.10.10.10 remote-as 100	Specify the neighbor's IPv4 address (10.10.10.10) and Autonomous System (AS) number (100). Router 1 is a neighbor of Router 2 .
awplus(config-router-af)# neighbor 10.10.10.10 activate	Activate the exchange of information to the peer (neighbor) with the IPv4 address 10.10.10.10 (Router 1) in IPv4 Address Family Configuration mode.
awplus(config-router-af)# neighbor 10.10.10.10 weight 500	Add a weight of 500 to all IPv4 routes coming from the neighbor with IPv4 address 10.10.10.10 (Router 1).
awplus(config-router-af)# exit-address-family	Exit IPv4 Address Family Configuration mode, and return to the Router Configuration mode.
awplus(config-router)# address-family ipv6 unicast	Enter the IPv6 Address Family Configuration mode from the Router Configuration mode.
awplus(config-router-af)# neighbor 2001:db8::116 remote-as 300	Specify the neighbor's IPv4 address (2001:db8::116) (and Autonomous System (AS) number (300)). Router 3 is a neighbor of Router 2 .
awplus(config-router-af)# neighbor 2001:db8::116 activate	Activate the exchange of information to the peer (neighbor) with the IPv6 address 2001:db8::116 (Router 3) in IPv6 Address Family Configuration mode. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
awplus(config-router-af)# neighbor 2001:db8::116 weight 700	Add a weight of 500 to all IPv6 routes coming from the neighbor with IPv6 address 2001:db8::116 (Router 3).
awplus(config-router-af)# exit-address-family	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
awplus(config-router)# exit	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 31

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 300</code>	Define the BGP routing process. The number 300 specifies the AS number of Router 3 .
<code>awplus(config-router)# address-family ipv4 unicast</code>	Enter the IPv4 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)# neighbor 2001:db8::115 remote-as 200</code>	Specify the neighbor's IPv6 address (2001:db8::115) and Autonomous System (AS) number (200). Router 2 is the neighbor of Router 1 .
<code>awplus(config-router-af)# neighbor 2001:db8::115 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv6 address 2001:db8::115 for Router 2 in IPv6 Address Family Configuration mode. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)# exit-address-family</code>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)# exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

address-family (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor activate (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
neighbor weight (BGP and BGP4+)
router bgp (BGP and BGP4+)

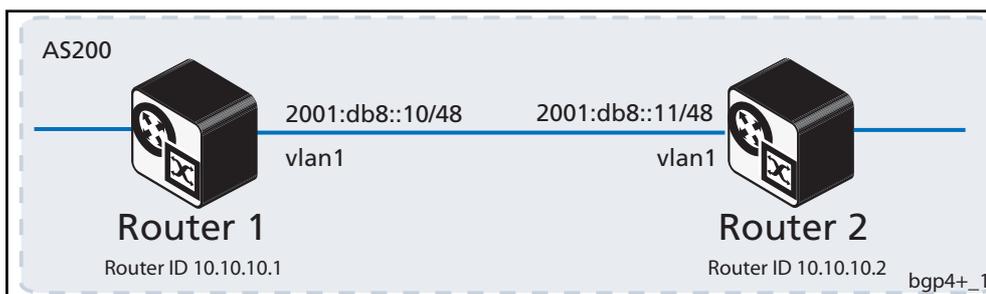
Validation Commands

show ip bgp (BGP)
show ip bgp neighbors (BGP)
show ip bgp summary (BGP)

BGP4+ Configurations

Enabling iBGP Peering using a Global Address

This example shows the minimum configuration required for enabling iBGP on a VLAN interface. **Router 1** and **Router 2** are BGP4+ Peers in the same Autonomous System (AS), 200, connected to network `2001:db8::/48`. First, specify an IPv6 global address, then define the AS number for the routers. Next, configure a Router ID, then define BGP4+ neighbors to start exchanging routing updates.



Router 1

```

awplus#
configure terminal Enter the Global Configuration mode.

awplus(config)#
interface vlan1 Specify the interface (vlan1) to configure an IPv6 global address.

awplus(config-if)#
ipv6 address 2001:db8::10/48 Specify the IPv6 global address 2001:db8::10/48 on interface
vlan1.

awplus(config-if)#
exit Leave Interface Configuration mode and enter Global
Configuration mode.

awplus(config)#
router bgp 200 Define the BGP4+ routing process. The number 200 specifies the
AS number of Router 1.

awplus(config-router)#
bgp router-id 10.10.10.1 Configure a fixed Router ID (10.10.10.1) for the BGP4+ routing
process.

```

<pre>awplus(config-router)# neighbor 2001:db8::11 remote- as 200</pre>	<p>Define BGP4+ neighbor Router 2, and establish a TCP session by specifying the global unicast IPv6 address (2001:db8::11) and the AS number (200) of neighbor Router 2.</p>
<pre>awplus(config-router)# address-family ipv6</pre>	<p>Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::11 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) Router 2 (2001:db8::11) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Router 2

<pre>awplus# configure terminal</pre>	<p>Enter the Global Configuration mode.</p>
<pre>awplus(config)# interface vlan1</pre>	<p>Specify the interface (vlan1) to configure an IPv6 global address.</p>
<pre>awplus(config-if)# ipv6 address 2001:db8::11/48</pre>	<p>Specify the IPv6 global unicast address 2001:db8::10/48 on interface vlan1.</p>
<pre>awplus(config-if)# exit</pre>	<p>Leave Interface Configuration mode and enter Global Configuration mode.</p>
<pre>awplus(config)# router bgp 200</pre>	<p>Define the BGP4+ routing process. The number 200 specifies the AS number of Router 2.</p>
<pre>awplus(config-router)# bgp router-id 10.10.10.2</pre>	<p>Configure a fixed Router ID (2001:db8::10) for the BGP4+ routing process.</p>

<pre>awplus(config-router)# neighbor 2001:db8::10 remote-as 200</pre>	<p>Define BGP4+ neighbor Router 1, and establish a TCP session by specifying the global IPv6 address (2001:db8::10) and the AS number (200) of neighbor Router 1.</p>
<pre>awplus(config-router)# address-family ipv6</pre>	<p>Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::10 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) Router 1 (2001:db8::10) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Names of Commands Used

address-family (BGP and BGP4+)
bgp router-id (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

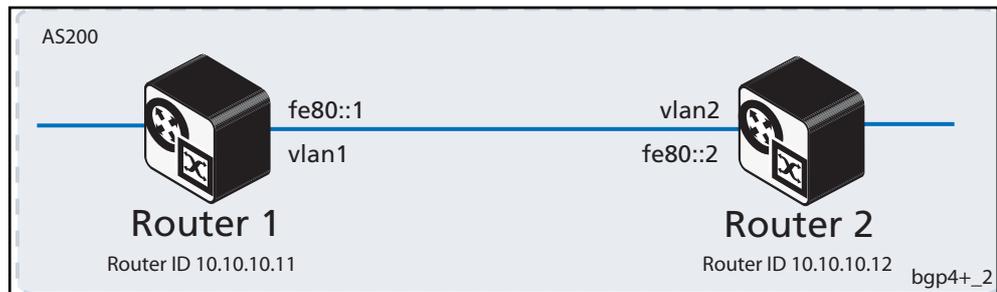
Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Enabling iBGP Peering using a Link-local Address

This example shows the minimum configuration required for enabling iBGP on a VLAN interface. **Router 1** and **Router 2** are BGP4+ Peers in the same Autonomous System (AS), 200, connected to link local network `fe80::/10`.

First, define the BGP4+ routing process, then define the AS number for the routers. Next, configure a Router ID for the BGP4+ routing process, then define BGP4+ neighbors to start exchanging routing updates.



Router 1

<code>awplus#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config)#</code>	
<code>router bgp 200</code>	Define the BGP4+ routing process. The number 200 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>bgp router-id 10.10.10.11</code>	Configure a fixed Router ID (10.10.10.11) for the BGP4+ routing process.
<code>awplus(config-router)#</code>	
<code>neighbor fe80::2 remote-as 200</code>	Define BGP4+ neighbor Router 2 , and establish a TCP session by specifying the link-local IPv6 address (<code>fe80::2</code>) and the AS number (200) of neighbor Router 2 .

<pre>awplus(config-router)# neighbor fe80::2 interface vlan1</pre>	<p>Specify a link-local neighbor, and configure the interface name of the neighbor (fe80::2).</p>
<pre>awplus(config-router)# address-family ipv6</pre>	<p>Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.</p>
<pre>awplus(config-router-af)# neighbor fe80:2 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) Router 2 (fe80::2) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Router 2

<pre>awplus# configure terminal</pre>	<p>Enter the Global Configuration mode.</p>
<pre>awplus(config)# router bgp 200</pre>	<p>Define the BGP4+ routing process. The number 200 specifies the AS number of Router 2.</p>
<pre>awplus(config-router)# bgp router-id 10.10.10.12</pre>	<p>Configure a fixed Router ID (10.10.10.12) for the BGP4+ routing process.</p>
<pre>awplus(config-router)# neighbor fe80::1 remote-as 200</pre>	<p>Define BGP4+ neighbor Router 1, and establish a TCP session by specifying the link-local IPv6 address (fe80::1) and the AS number (200) of neighbor Router 1.</p>

<pre>awplus(config-router)# neighbor fe80::1 interface vlan2</pre>	Specify a link-local neighbor, and configure the interface name of the neighbor (fe80::1).
<pre>awplus(config-router)# address-family ipv6</pre>	Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.
<pre>awplus(config-router-af)# neighbor fe80::1 activate</pre>	Activate the exchange of information to the peer (neighbor) Router 1 (fe80::1) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<pre>awplus(config-router-af)# exit-address-family</pre>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<pre>awplus(config-router)# exit</pre>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

address-family (BGP and BGP4+)
bgp router-id (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor activate (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Enabling eBGP Peering between different Autonomous Systems

This example shows the minimum configuration required to enable eBGP on a VLAN interface, when the routers belong to different Autonomous Systems. **Router 1** and **Router 2** are two routers in different Autonomous Systems, 200 and 300, connecting to network 2001:db8::/64.



Router 1

<code>awplus#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config)#</code>	
<code>router bgp 200</code>	Define the BGP4+ routing process. The number 200 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>bgp router-id 10.10.10.11</code>	Configure a fixed Router ID (10.10.10.11) for the BGP4+ routing process.
<code>awplus(config-router)#</code>	
<code>neighbor 2001:db8::80:2</code> <code>remote-as 300</code>	Define BGP4+ neighbor Router 2 , and establish a TCP session by specifying the global unicast IPv6 address (2001:db8::80:2) and the AS number (300) of neighbor Router 2 .
<code>awplus(config-router)#</code>	
<code>address-family ipv6</code>	Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.
<code>awplus(config-router-af)#</code>	
<code>neighbor 2001:db8::80:2</code> <code>activate</code>	Activate the exchange of information to the peer (neighbor) Router 2 (2001:db8::80:2) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)#</code>	
<code>exit-address-family</code>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)#</code>	
<code>exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 2

<code>awplus#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config)#</code>	
<code>router bgp 300</code>	Define the BGP4+ routing process. The number 300 specifies the AS number of Router 2 .
<code>awplus(config-router)#</code>	
<code>bgp router-id 10.10.10.12</code>	Configure a fixed Router ID (10.10.10.12) for the BGP4+ routing process.
<code>awplus(config-router)#</code>	
<code>neighbor 2001:db8::80:1 remote-as 200</code>	Define BGP4+ neighbor Router 1 , and establish a TCP session by specifying the global unicast IPv6 address (2001:db8::80:1) and the AS number (200) of neighbor Router 1 .
<code>awplus(config-router)#</code>	
<code>address-family ipv6</code>	Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.
<code>awplus(config-router-af)#</code>	
<code>neighbor 2001:db8::80:1 activate</code>	Activate the exchange of information to the peer (neighbor) Router 1 (2001:db8::80:1) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)#</code>	
<code>exit-address-family</code>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)#</code>	
<code>exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

address-family (BGP and BGP4+)
bgp router-id (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor activate (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

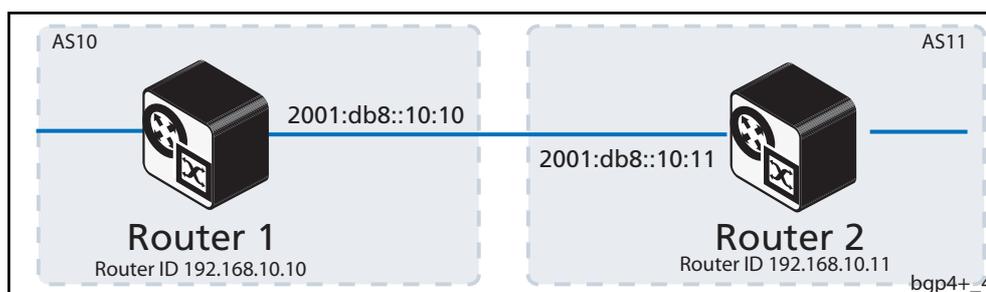
Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Configuring Route-Maps with BGP4+

Use route-maps to filter incoming updates from a BGP4+ peer. In this example, a prefix-list named `list1` on **Router 1** is configured to deny entry of any routes with the IPv6 address `2001::db8:12::/48`. To test the route-map filter, **Router 2** is configured to advertise IPv6 network prefixes `2001:db8:11::/48` and `2001:db8:12::/48`.

Use the `show bgp ipv6` command on **Router 1** to verify only the `2001:db8:11::/48` network prefix has been installed in the BGP table.



Router 1

```
awplus#
configure terminal
```

Enter the Global Configuration mode.

```
awplus(config)#
ipv6 prefix-list FILTER_LIST_IN
seq 5 deny 2001:db8:12::/48
```

Create an entry in the prefix-list named `FILTER_LIST_IN` for the route-map. `5` specifies the sequence number or position of this specific route-map. `deny` specifies that prefixes will be rejected. `permit` specifies that prefixes will be allowed. `48` and `64` are the minimum and maximum prefix lengths, respectively, to be matched for this route-map. The prefix-list permits or denies selected prefixes from being installed in the BGP table, when used in a route map with the route map applied to the peering.

```
awplus(config)#
ipv6 prefix-list FILTER_LIST_IN
seq 10 permit any
```

Create another entry in the prefix-list named `FILTER_LIST_IN` for the route-map. `10` specifies the sequence number or position of this specific route-map. `permit any` specifies prefix-list `FILTER_LIST_IN` to accept all prefixes of any length. The prefix-list permits or denies selected prefixes from being installed in the BGP table, when used in a route map with the route map applied to the peering.

```
awplus(config)#
route-map FILTER_BGP_IN_MAP
permit 1
```

Enter the Route-map Configuration mode from the Global Configuration mode and creates the route-map named `FILTER_BGP_IN_MAP`. The route-map matches on the prefix-list.

```
awplus(config-route-map)#
match ipv6 address prefix-list
FILTER_LIST_IN
```

Set the match criteria for the route-map. In this case, if the route-map name matches `FILTER_LIST_IN`, then the first sequence `5` denies prefixes and the second sequence `10` accepts prefixes.

awplus(config-router-map)#	
exit	Exit the Route-map Configuration mode and return to the Global Configuration mode.
awplus(config)#	
router bgp 10	Define the routing process, where 10 is the Autonomous System (AS) number for Router 1 .
awplus(config-router)#	
bgp router-id 192.168.10.10	Configure a fixed Router ID (192.168.10.10) for the BGP4+ routing process.
awplus(config-router)#	
neighbor 2001:db8::10:11 remote-as 11	Define BGP neighbor Router 2 , and establish a TCP session by specifying the global unicast IPv6 address (2001:db8::10:11) and the AS number (11) of neighbor Router 2 .
awplus(config-router)#	
address-family ipv6	Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.
awplus(config-router-af)#	
neighbor 2001:db8::10:11 route-map list1 in	Apply the route-map list1 to all incoming routes.
awplus(config-router-af)#	
neighbor 2001:db8::10:11 activate	Activate the neighbor Router 2 (2001:db8::10:11), and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
awplus(config-router-af)#	
exit-address-family	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
awplus(config-router)#	
exit	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 2

awplus# configure terminal	Enter the Global Configuration mode.
awplus(config)# router bgp 11	Define the BGP4+ routing process. The number 11 specifies the Autonomous System (AS) number of Router 2 .
awplus(config-router)# bgp router-id 192.168.10.11	Configure a fixed Router ID (192.168.10.11) for the BGP4+ routing process.
awplus(config-router)# neighbor 2001:db8::10:10 remote-as 10	Define BGP4+ neighbor Router 1 , and establish a TCP session by specifying the global unicast IPv6 address (2001:db8::10:10) and the AS number (10) of neighbor Router 1 .
awplus(config-router)# address-family ipv6	Enter IPv6 Address Family Configuration mode for configuring routing sessions that use IPv6 address prefixes.
awplus(config-router-af)# network 2001:db8:11::/48	Announces IPv6 network prefix 2001:db8:11::/48. Running <code>show ipv6 bgp</code> command on Router 1 displays Router 1 receiving 2001:db8:11::/48 packets.
awplus(config-router-af)# network 2001:db8:12::/48	Announces IPv6 network prefix 2001:db8:12::/48. Running <code>show ipv6 bgp</code> command on Router 1 displays Router 1 rejecting 2001:db8:12::/48 packets.
awplus(config-router-af)# neighbor 2001:db8::10:10 activate	Activate the exchange of information to the peer (neighbor) Router 1 (2001:db8::10:10) and enable the exchange of IPv6 address prefix types with this neighbor. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
awplus(config-router-af)# exit-address-family	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
awplus(config-router)# exit	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

address-family (BGP and BGP4+)
bgp router-id (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
network (BGP and BGP4+)
neighbor activate (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
neighbor route-map (BGP and BGP4+)
router bgp (BGP and BGP4+)

Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Configuring BGP4+ Graceful Restart

Using BGP4+ graceful restart, the data forwarding plane of the device can continue to process and forward packets even if the control plane, which is responsible for determining best paths, fails.

The **bgp graceful-restart (BGP and BGP4+)** command on page 3.32 sets the maximum time that a graceful-restart neighbor waits to come back up after a restart. This value is applied to all neighbors unless you explicitly override it by configuring the corresponding value on the neighbor.

This example shows configuring BGP4+ graceful restart.



Router 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP4+ routing process. The number 100 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>bgp router-id 192.168.10.10</code>	Configure a fixed Router ID (192.168.10.10) for the BGP4+ routing process.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart</code>	Enable BGP4+ graceful restart support. If the restart and stale-path timers are not configured, the defaults are 90 seconds and 360 seconds, respectively.
<code>awplus(config-router)#</code>	
<code>neighbor 2001:db8::10:11 remote-as 200</code>	Specify the neighbor's IPv6 address (2001:db8::10:11) and the Autonomous System (AS) number of the neighbor (200) Router 2 . Router 2 is the neighbor of Router 1 . This command defines BGP neighbor Router 2 , and establishes a TCP session by specifying the global unicast IPv6 address and AS number of the neighbor.
<code>awplus(config-router)#</code>	
<code>address-family ipv6 unicast</code>	Exchange the IPv6 capabilities, and move from Router Configuration mode to IPv6 Address Family Configuration mode.
<code>awplus(config-router-af)#</code>	
<code>neighbor 2001:db8::10:11 activate</code>	Activate the exchange of information to the peer (neighbor) with the IPv6 address (2001:db8::10:11) (Router 2). Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)#</code>	
<code>neighbor 2001:db8::10:11 capability graceful-restart</code>	Specify the neighbor's IPv6 address (2001:db8::10:11), for which the graceful restart capability function is supported. This command advertises the graceful restart capability to the peer.
<code>awplus(config-router-af)#</code>	
<code>exit-address-family</code>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)#</code>	
<code>exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 2

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 200</code>	Define the BGP4+ routing process. The number 200 specifies the AS number of Router 2 .
<code>awplus(config-router)# bgp router-id 192.168.10.11</code>	Configure a fixed Router ID (192.168.10.11) for the BGP4+ routing process.
<code>awplus(config-router)# bgp graceful-restart</code>	Enable BGP4+ graceful restart support.
<code>awplus(config-router)# bgp graceful-restart restart-time 120</code>	Configure the maximum time in seconds (120) required for neighbor(s) to restart.
<code>awplus(config-router)# bgp graceful-restart stalepath-time 120</code>	Configure the maximum time in seconds (120) to retain stale paths from the restarting neighbor(s).
<code>awplus(config-router)# bgp update-delay 120</code>	Configure the maximum time in seconds (120) to defer initial route selection.
<code>awplus(config-router)# neighbor 2001:db8::10:10 remote-as 100</code>	Specify the neighbor's global unicast IPv6 address (2001:db8::10:10) and the AS number of the neighbor (100) Router 1 . Router 1 is the neighbor of Router 2 . This command defines BGP neighbor Router 1 , and establishes a TCP session by specifying the global unicast IPv6 address and AS number of the neighbor.

<pre>awplus(config-router)# address-family ipv6 unicast</pre>	<p>Exchange the IPv6 capabilities, and move from Router Configuration mode to IPv6 Address Family Configuration mode.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::10:10 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address (2001:db8::10:10) (Router 1). Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::10:10 capability graceful-restart</pre>	<p>Specify the neighbor's global unicast IPv6 address (2001:db8::10:10), for which the graceful restart capability function is supported. This command advertises the graceful restart capability to the peer.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Names of Commands Used

address-family (BGP and BGP4+)
bgp graceful-restart (BGP and BGP4+)
bgp graceful-restart graceful-reset (BGP and BGP4+)
bgp update-delay (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor capability graceful-restart (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

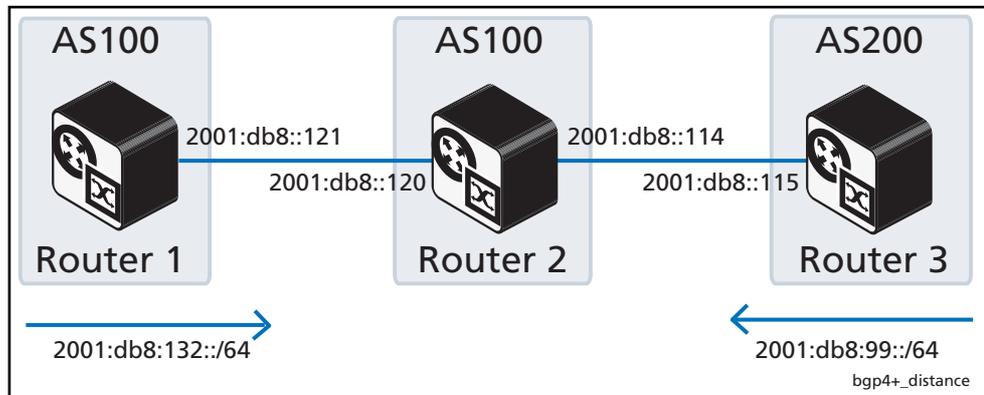
Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Configuring BGP4+ Distance

You can configure administrative distance in BGP4+ for a specific address family.

This example shows configuring the BGP4+ distance for the IPv6 Address Family.



Router 1

```
awplus(config)#
configure terminal Enter the Global Configuration mode.

awplus(config-router)#
router bgp 100 Define the BGP4+ routing process. The number 100
                specifies the AS number of Router 1.

awplus(config-router)#
network 2001:db8:132::/64 Specify the network 2001:db8:132::/64 to be
                          advertised by the BGP4+ routing process from Router 1
                          to Router 2.
```

<pre>awplus(config-router)# neighbor 2001:db8::120 remote-as 100</pre>	<p>Specify the neighbor's IPv6 address (2001:db::120) and Autonomous System (AS) number (100). Router 2 is the neighbor of Router 1. This command defines BGP neighbor Router 2, and establishes a TCP session by specifying the global unicast IPv6 address and the AS number of the neighbor.</p>
<pre>awplus(config-router)# address-family ipv6 unicast</pre>	<p>Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::120 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address 2001:db::120 for Router 2. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Router 2

<pre>awplus(config)# configure terminal</pre>	<p>Enter the Global Configuration mode.</p>
<pre>awplus(config-router)# router bgp 100</pre>	<p>Define the BGP4+ routing process. The number 100 specifies the AS number of Router 2.</p>
<pre>awplus(config-router)# neighbor 2001:db8::121 remote-as 100</pre>	<p>Specify the neighbor's global unicast IPv6 address (2001:db8::121) and Autonomous System (AS) number (100). Router 1 is a neighbor of Router 2. This command defines BGP4+ neighbor Router 1, and establishes a TCP session by specifying the global unicast IPv6 address and the AS number of the neighbor.</p>
<pre>awplus(config-router)# neighbor 2001:db8::115 remote-as 200</pre>	<p>Specify the neighbor's global unicast IPv6 address (2001:db8::115) (and Autonomous System (AS) number (200). Router 3 is a neighbor of Router 2. This command defines BGP4+ neighbor Router 3, and establishes a TCP session by specifying the global unicast IPv6 address and the AS number of the neighbor.</p>
<pre>awplus(config-router)# address-family ipv6 unicast</pre>	<p>Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.</p>

<pre>awplus(config-router-af)# aggregate-address 2001:db8::102/64 summary-only</pre>	<p>Configure an IPv6 non-AS-set aggregate route on Router 2. Note that the local distance gets applied to this route.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::121 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address 2001:db8::121 for Router 1. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::115 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address 2001:db8::115 for Router 3. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# distance 12 13 11</pre>	<p>Configure the administrative distance for external, internal, and local routes received in IPv6 Address Family Configuration mode.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Router 3

<code>awplus(config)# configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)# router bgp 200</code>	Define the BGP4+ routing process. The number 100 specifies the AS number of Router 3 .
<code>awplus(config-router)# network 2001:db8:99::/64</code>	Specify the network 2001:db8:99::/64 to be advertised by the BGP4+ routing process from Router 3 to Router 2 .
<code>awplus(config-router)# neighbor 2001:db8::114 remote-as 100</code>	Specify the neighbor's global unicast IPv6 address (2001:db8::114) (and Autonomous System (AS) number (100). Router 2 is a neighbor of Router 3 . This command defines BGP4+ neighbor Router 2 , and establishes a TCP session by specifying the global unicast IPv6 address and the AS number of the neighbor.
<code>awplus(config-router)# address-family ipv6 unicast</code>	Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.
<code>awplus(config-router-af)# neighbor 2001:db8::114 activate</code>	Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address 2001:db8::114 for Router 2 . Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<code>awplus(config-router-af)# exit-address-family</code>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<code>awplus(config-router)# exit</code>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Names of Commands Used

address-family (BGP and BGP4+)
aggregate-address (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
network (BGP and BGP4+)
neighbor activate (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

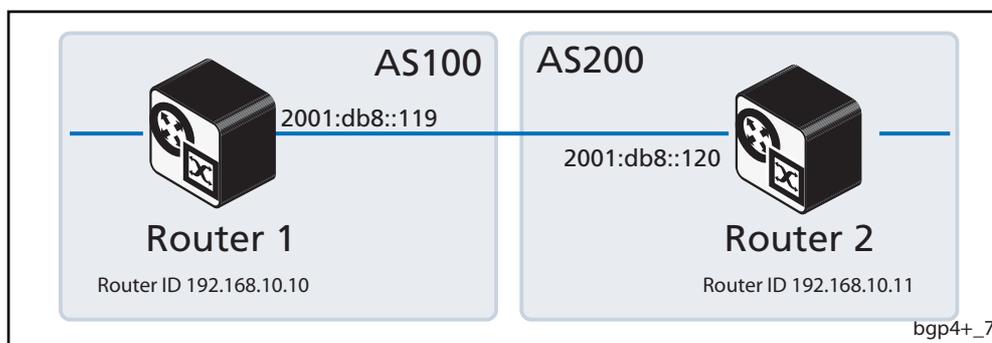
Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Configuring BGP4+ Graceful Reset

The graceful restart feature for BGP4+ session reset is used so that any changes in network configuration do not affect packet forwarding. The **bgp graceful-restart graceful-reset (BGP and BGP4+)** command on page 3.33 invokes graceful restart only when a configuration change forces a peer reset.

This example shows configuring BGP4+ graceful reset.



Router 1

<code>awplus(config)#</code>	
<code>configure terminal</code>	Enter the Global Configuration mode.
<code>awplus(config-router)#</code>	
<code>router bgp 100</code>	Define the BGP4+ routing process. The number 100 specifies the AS number of Router 1 .
<code>awplus(config-router)#</code>	
<code>bgp router-id 192.168.10.10</code>	Configure a fixed Router ID (192.168.10.10) for the BGP4+ routing process.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart</code>	Enable BGP4+ graceful restart support.
<code>awplus(config-router)#</code>	
<code>bgp graceful-restart graceful-reset</code>	Configure to invoke graceful restart when a configuration change forces a peer reset.
<code>awplus(config-router)#</code>	
<code>neighbor 2001:db8::120 remote-as 200</code>	Specify the neighbor's IPv6 address (2001:db8::120) and the AS number of the neighbor (200).

<pre>awplus(config-router)# address-family ipv6 unicast</pre>	Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.
<pre>awplus(config-router-af)# neighbor 2001:db8::120 activate</pre>	Activate the exchange of information to the peer (neighbor) with the IPv6 address 2001:db8::120 for Router 2 . Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.
<pre>awplus(config-router-af)# neighbor 2001:db8::120 capability graceful-restart</pre>	Specify the neighbor's global unicast IPv6 address (2001:db8::120) for which the graceful restart capability is supported. This command advertises the graceful restart capability to the peer.
<pre>awplus(config-router-af)# exit-address-family</pre>	Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.
<pre>awplus(config-router)# exit</pre>	Exit the Router Configuration mode, and return to the Global Configuration mode.

Router 2

<pre>awplus(config)# configure terminal</pre>	Enter the Global Configuration mode.
<pre>awplus(config-router)# router bgp 200</pre>	Define the BGP4+ routing process. The number 200 specifies the AS number of Router 2 .
<pre>awplus(config-router)# bgp router-id 192.168.10.10</pre>	Configure a fixed Router ID (192.168.10.10) for the BGP4+ routing process.
<pre>awplus(config-router)# bgp graceful-restart</pre>	Enable BGP4+ graceful restart support.
<pre>awplus(config-router)# bgp graceful-restart graceful-reset</pre>	Configure to invoke graceful restart when a configuration change forces a peer reset.
<pre>awplus(config-router)# neighbor 2001:db8::119 remote-as 100</pre>	Specify the neighbor's IPv6 address (2001:db8::119) and the AS number of the neighbor (100).

<pre>awplus(config-router)# address-family ipv6 unicast</pre>	<p>Enter the IPv6 Address Family Configuration mode from the Global Configuration mode.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::119 activate</pre>	<p>Activate the exchange of information to the peer (neighbor) with the global unicast IPv6 address 2001:db8::119 for Router 1. Note that other information apart from prefixes is sent in updates. This command activates the exchange of information for the peer defined.</p>
<pre>awplus(config-router-af)# neighbor 2001:db8::119 capability graceful-restart</pre>	<p>Specify the neighbor's global unicast IPv6 address (2001:db8::119) for which the graceful restart capability is supported. This command advertises the graceful restart capability to the peer.</p>
<pre>awplus(config-router-af)# exit-address-family</pre>	<p>Exit IPv6 Address Family Configuration mode, and return to the Router Configuration mode.</p>
<pre>awplus(config-router)# exit</pre>	<p>Exit the Router Configuration mode, and return to the Global Configuration mode.</p>

Names of Commands Used

address-family (BGP and BGP4+)
bgp graceful-restart (BGP and BGP4+)
bgp graceful-restart graceful-reset (BGP and BGP4+)
exit-address-family (BGP and BGP4+)
neighbor capability graceful-restart (BGP and BGP4+)
neighbor remote-as (BGP and BGP4+)
router bgp (BGP and BGP4+)

Validation Commands

show bgp ipv6 (BGP4+)
show bgp ipv6 neighbors (BGP4+)
show bgp ipv6 summary (BGP4+)

Chapter 3: BGP and BGP4+ Commands



Introduction	3.5
Controlling "show" Command Output.....	3.6
Command List	3.8
address-family (BGP and BGP4+).....	3.8
aggregate-address (BGP and BGP4+)	3.10
auto-summary (BGP).....	3.12
bgp aggregate-nexthop-check (BGP and BGP4+)	3.13
bgp always-compare-med (BGP and BGP4+)	3.14
bgp bestpath as-path ignore (BGP and BGP4+)	3.15
bgp bestpath compare-confed-aspath (BGP and BGP4+)	3.15
bgp bestpath compare-routerid (BGP and BGP4+)	3.16
bgp bestpath med (BGP and BGP4+)	3.17
bgp bestpath med remove-recv-med (BGP and BGP4+)	3.18
bgp bestpath med remove-send-med (BGP and BGP4+)	3.18
bgp client-to-client reflection (BGP and BGP4+)	3.19
bgp cluster-id (BGP and BGP4+).....	3.20
bgp confederation identifier (BGP and BGP4+)	3.21
bgp confederation peers (BGP and BGP4+)	3.22
bgp config-type (BGP and BGP4+)	3.24
bgp dampening (BGP and BGP4+)	3.26
bgp default ipv4-unicast (BGP and BGP4+).....	3.27
bgp default local-preference (BGP)	3.28
bgp deterministic-med (BGP and BGP4+).....	3.29
bgp enforce-first-as (BGP and BGP4+).....	3.30
bgp fast-external-failover (BGP and BGP4+)	3.31
bgp graceful-restart (BGP and BGP4+).....	3.32
bgp graceful-restart graceful-reset (BGP and BGP4+)	3.33
bgp log-neighbor-changes (BGP and BGP4+).....	3.34
bgp memory maxallocation (BGP and BGP4+)	3.36
bgp nexthop-trigger-count (BGP and BGP4+)	3.37
bgp nexthop-trigger delay (BGP and BGP4+).....	3.38
bgp nexthop-trigger enable (BGP and BGP4+).....	3.39
bgp rfc1771-path-select (BGP)	3.40
bgp rfc1771-strict (BGP).....	3.40
bgp router-id (BGP and BGP4+).....	3.41
bgp scan-time (BGP)	3.42
bgp update-delay (BGP and BGP4+).....	3.43
clear bgp * (BGP and BGP4+)	3.44
clear bgp (IPv4 or IPv6 address) (BGP and BGP4+)	3.45
clear bgp (ASN) (BGP and BGP4+).....	3.47
clear bgp external (BGP and BGP4+).....	3.48
clear bgp peer-group (BGP and BGP4+)	3.49
clear ip bgp * (BGP).....	3.50
clear ip bgp (IPv4) (BGP)	3.52
clear ip bgp dampening (BGP)	3.53
clear ip bgp flap-statistics (BGP)	3.53
clear ip bgp (ASN) (BGP)	3.54

clear ip bgp external (BGP)	3.55
clear ip bgp peer-group (BGP).....	3.56
clear bgp ipv6 (ipv6 address) (BGP4+)	3.57
clear bgp ipv6 dampening (BGP4+)	3.58
clear bgp ipv6 flap-statistics (BGP4+)	3.58
clear bgp ipv6 (ASN) (BGP4+)	3.59
clear bgp ipv6 external (BGP4+)	3.60
clear bgp ipv6 peer-group (BGP4+)	3.61
debug bgp (BGP)	3.62
distance (BGP and BGP4+).....	3.63
exit-address-family (BGP and BGP4+).....	3.65
ip as-path access-list (BGP and BGP4+).....	3.66
ip community-list (BGP and BGP4+)	3.67
ip community-list expanded (BGP and BGP4+)	3.68
ip community-list standard (BGP and BGP4+).....	3.70
ip extcommunity-list expanded (BGP and BGP4+).....	3.72
ip extcommunity-list standard (BGP and BGP4+)	3.74
ip prefix-list (IPv4 Prefix List)	3.76
ipv6 prefix-list (IPv6 Prefix List)	3.78
match as-path (Route Map).....	3.80
match community (Route Map).....	3.81
neighbor activate (BGP and BGP4+)	3.82
neighbor advertisement-interval (BGP and BGP4+)	3.85
neighbor allowas-in (BGP and BGP4+).....	3.88
neighbor as-origination-interval (BGP and BGP4+).....	3.91
neighbor attribute-unchanged (BGP and BGP4+)	3.93
neighbor capability graceful-restart (BGP and BGP4+).....	3.96
neighbor capability orf prefix-list (BGP and BGP4+)	3.99
neighbor capability route-refresh (BGP and BGP4+)	3.102
neighbor collide-established (BGP and BGP4+)	3.105
neighbor default-originate (BGP and BGP4+)	3.107
neighbor description (BGP and BGP4+)	3.110
neighbor disallow-infinite-holdtime (BGP and BGP4+)	3.112
neighbor distribute-list (BGP and BGP4+)	3.114
neighbor dont-capability-negotiate (BGP and BGP4+)	3.117
neighbor ebgp-multihop (BGP and BGP4+)	3.119
neighbor enforce-multihop (BGP and BGP4+).....	3.122
neighbor filter-list (BGP and BGP4+).....	3.125
neighbor interface (BGP4+).....	3.128
neighbor local-as (BGP and BGP4+)	3.129
neighbor maximum-prefix (BGP and BGP4+).....	3.131
neighbor next-hop-self (BGP and BGP4+)	3.134
neighbor override-capability (BGP and BGP4+)	3.137
neighbor passive (BGP and BGP4+)	3.139
neighbor password (BGP and BGP4+).....	3.141
neighbor peer-group (add a neighbor) (BGP and BGP+)	3.144
neighbor peer-group (create a peer-group) (BGP and BGP4+).....	3.146
neighbor port (BGP and BGP4+).....	3.147
neighbor prefix-list (BGP and BGP4+)	3.149
neighbor remote-as (BGP and BGP4+).....	3.152
neighbor remove-private-AS (BGP).....	3.155
neighbor restart-time (BGP and BGP4+).....	3.156
neighbor route-map (BGP and BGP4+).....	3.159
neighbor route-reflector-client (BGP).....	3.164
neighbor route-server-client (BGP)	3.165
neighbor send-community (BGP and BGP4+).....	3.166
neighbor shutdown (BGP and BGP4+).....	3.170

neighbor soft-reconfiguration inbound (BGP and BGP4+)	3.172
neighbor timers (BGP and BGP4+).....	3.175
neighbor transparent-as (BGP and BGP4+)	3.178
neighbor transparent-next-hop (BGP and BGP4+)	3.180
neighbor unsuppress-map (BGP and BGP4+)	3.182
neighbor update-source (BGP and BGP4+).....	3.185
neighbor version (BGP)	3.188
neighbor weight (BGP and BGP4+)	3.190
network (BGP and BGP4+)	3.193
network synchronization (BGP and BGP4+).....	3.196
redistribute (into BGP or BGP4+) (BGP and BGP4+)	3.197
restart bgp graceful (BGP).....	3.199
router bgp (BGP and BGP4+)	3.200
route-map (Route Map)	3.201
set as-path (Route Map).....	3.203
set community (Route Map)	3.204
show bgp ipv6 (BGP4+)	3.205
show bgp ipv6 community (BGP4+).....	3.206
show bgp ipv6 community-list (BGP4+).....	3.207
show bgp ipv6 dampening (BGP4+).....	3.208
show bgp ipv6 filter-list (BGP4+).....	3.209
show bgp ipv6 inconsistent-as (BGP4+).....	3.209
show bgp ipv6 longer-prefixes (BGP4+).....	3.210
show bgp ipv6 neighbors (BGP4+).....	3.211
show bgp ipv6 paths (BGP4+).....	3.214
show bgp ipv6 prefix-list (BGP4+)	3.214
show bgp ipv6 quote-regexp (BGP4+).....	3.215
show bgp ipv6 regexp (BGP4+)	3.216
show bgp ipv6 route-map (BGP4+).....	3.217
show bgp ipv6 summary (BGP4+)	3.217
show bgp next-hop-tracking (BGP)	3.218
show bgp next-hop-tree-details (BGP).....	3.218
show bgp memory maxallocation (BGP).....	3.218
show debugging bgp (BGP).....	3.219
show ip bgp (BGP)	3.220
show ip bgp attribute-info (BGP)	3.221
show ip bgp cidr-only (BGP)	3.222
show ip bgp community (BGP)	3.223
show ip bgp community-info (BGP).....	3.224
show ip bgp community-list (BGP).....	3.225
show ip bgp dampening (BGP).....	3.226
show ip bgp filter-list (BGP).....	3.228
show ip bgp inconsistent-as (BGP).....	3.229
show ip bgp longer-prefixes (BGP).....	3.230
show ip bgp neighbors (BGP).....	3.231
show ip bgp neighbors connection-retrytime (BGP).....	3.234
show ip bgp neighbors hold-time (BGP).....	3.234
show ip bgp neighbors keepalive (BGP).....	3.235
show ip bgp neighbors keepalive-interval (BGP).....	3.235
show ip bgp neighbors notification (BGP).....	3.236
show ip bgp neighbors open (BGP)	3.236
show ip bgp neighbors rcvd-msgs (BGP)	3.237
show ip bgp neighbors sent-msgs (BGP)	3.237
show ip bgp neighbors update (BGP)	3.238
show ip bgp paths (BGP)	3.238
show ip bgp prefix-list (BGP).....	3.239
show ip bgp quote-regexp (BGP).....	3.240

show ip bgp regexp (BGP)	3.241
show ip bgp route-map (BGP)	3.242
show ip bgp scan (BGP)	3.243
show ip bgp summary (BGP).....	3.244
show ip community-list (BGP and BGP4+).....	3.245
show ip extcommunity-list (BGP and BGP4+)	3.245
show ip prefix-list (IPv4 Prefix List)	3.246
show ipv6 prefix-list (IPv6 Prefix List)	3.246
show ip protocols bgp (BGP)	3.247
show route-map (Route Map)	3.248
synchronization (BGP and BGP4+).....	3.249
timers (BGP and BGP4+)	3.250
undebug bgp (BGP)	3.251

Introduction

This chapter provides an alphabetical reference of commands used to configure the Border Gateway Protocol for IPv4 (BGP) and for IPv6 (BGP4+).

For basic BGP and BGP4+ introduction information, including terminology and concepts about BGP and BGP4+, including the concept of autonomous system numbers (ASN), path attributes (PA), and both internal and external BGP/BGP4+, see [Chapter 1, BGP and BGP4+ Introduction](#).

For basic BGP and BGP4+ configuration examples, see [Chapter 2, BGP and BGP4+ Configuration](#).

Controlling “show” Command Output

You can control the output of **show** commands by using the | and > or >> tokens in the following ways:

- To display only part of the output, follow the command with | and then other keywords (see **Output Modifiers** below)
- To save the output to a file, follow the command with > **filename**
- To append the output to an existing file, follow the command with >> **filename**

Using the ? after typing the **show** command displays the following information about these tokens:

```
awplus# show users
```

```
| Output modifiers
> Output redirection
>> Output redirection (append)
```

Output Modifiers Type the | (vertical bar) to use **Output modifiers**.

```
append      Append output
begin       Begin with the first line that contains
            matching output
exclude     Exclude lines that contain matching output
include     Include lines that contain matching output
redirect    Redirect output
```

Begin The **begin** parameter causes the display to begin at the first line that contains the input string.

```
awplus# show run | begin vlan1
```

```
...skipping
interface vlan1
 ip address 192.168.14.1
 !!
line con 0
 login
line vty 0 4
 login
!
end
```

Exclude The **exclude** parameter excludes all lines of output that contain the input string. In the following output all lines containing the word "input" are excluded:

```
awplus# show interface vlan1 | exclude input
```

```
Interface vlan1
Scope: both
Hardware is Ethernet, address is 192.168.14.1
index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
Label switching is disabled
No Virtual Circuit configured
Administrative Group(s): None
DSTE Bandwidth Constraint Mode is MAM
  output packets 4438, bytes 394940, dropped 0
  output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0,
window 0
  collisions 0
```

Include The **include** parameter includes only those lines of output that contain the input string. In the output below, all lines containing the word "input" are included:

```
awplus# show interface vlan1 | include input
```

```
  input packets 80434552, bytes 2147483647, dropped 0, multicast
  packets 0
  input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 1,
  missed 0
```

Redirect The **redirect** parameter puts the lines of output into the specified file. If the file already exists, the new output overwrites the file's contents; the new output is not appended to the existing file contents.

| **redirect** and **>** are synonyms.

```
awplus# show history | redirect history.txt
```

Output Redirection The output redirection token **>** puts the lines of output into the specified file. If the file already exists, the new output overwrites the file's contents; the new output is not appended to the existing file contents.

| **redirect** and **>** are synonyms.

```
awplus# show history > history.txt
```

Append Output The append output token **>>** adds the lines of output into the specified file. The file must already exist, for the new output to be added to the end of the file's contents; the new output is appended to the existing file contents.

| **append** and **>>** are synonyms.

```
awplus# show history >> history.txt
```

Command List

address-family (BGP and BGP4+)

This command enters the IPv4 or IPv6 Address-Family Configuration command mode. In this mode you can configure address-family specific parameters. When using VRF-Lite, you can enter IPv4 Address Family Configuration mode for a specified VRF instance before configuring that instance.

Syntax [BGP] address-family ipv4 [unicast]
no address-family ipv4 [unicast]

Syntax [VRF] address-family ipv4 [unicast|vrf <vrf-name>]
no address-family ipv4 [unicast|vrf <vrf-name>]

Syntax [BGP4+] address-family ipv6 [unicast]
no address-family ipv6 [unicast]

Parameter	Description
ipv4	Configure parameters relating to the exchange of IPv4 prefixes.
ipv6	Configure parameters relating to the exchange of IPv6 prefixes.
unicast	Configure parameters relating to the exchange of routes to unicast destinations.
vrf	Applies the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance to enter IPv4 Address-Family mode for.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration mode for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] To leave the IPv4 or IPv6 Address Family Configuration mode, and return to the Router Configuration mode, use the **exit-address-family (BGP and BGP4+)** command.

Example [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# neighbor 192.168.0.1 remote-as 100
awplus(config-router)# address-family ipv4 vrf green
awplus(config-router-af)# neighbor 192.168.0.1 activate
awplus(config-router-af)# exit-address-family
awplus(config-router)#
```

Example [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 activate
awplus(config-router-af)# exit-address-family
awplus(config-router)#
```

Related Commands [exit-address-family \(BGP and BGP4+\)](#)

aggregate-address (BGP and BGP4+)

This command adds an aggregate route that can be advertised to BGP or BGP4+ neighbors. This command creates an aggregate entry in the BGP or BGP4+ routing table if the switch learns, by any means, any routes that are within the range configured by the aggregate address/mask.

When this command is used with the **summary-only** option, the more-specific routes of the aggregate are suppressed to all neighbors. Use the **neighbor unsuppress-map (BGP and BGP4+)** command instead to selectively leak more-specific routes to a particular neighbor.

The **no** variant of this command removes the aggregate configured by the **aggregate-address** command.

Syntax [BGP] `aggregate-address <ip-addr/m> {summary-only|as-set}`
`no aggregate-address <ip-addr/m> {summary-only|as-set}`

Syntax [BGP4+] `aggregate-address <ipv6-addr/prefix-length> {summary-only|as-set}`
`no aggregate-address <ipv6-addr/prefix-length> {summary-only|as-set}`

Parameter	Description
<code><ip-addr/m></code>	Specifies the aggregate IPv4 address and mask.
<code><ipv6-addr/prefix-length></code>	Specifies the aggregate IPv6 address. The IPv6 address uses the format X:X::X:X/Prefix-Length. The prefix-length is usually set between 0 and 64.
<code>summary-only</code>	Filters more specific routes from updates. Only the aggregate address/mask will be advertised, and none of the component addresses that fall within the range of the aggregate address/mask.
<code>as-set</code>	Generates AS set path information. The AS-path advertised with the aggregate is an unordered list of all the AS-numbers that appear in any of the AS-paths of the component routes, with each AS-number appearing just once in the list.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP] If the `summary-only` parameter is specified, then only the aggregate address/mask will be advertised, and none of the component addresses that fall within the range of the aggregate address/mask. For example, if you configure:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# aggregate-address 172.0.0.0/8 summary-only
```

then the switch will advertise the prefix 172.0.0.0/8, but no component routes like 172.10.0.0/16

The `as-set` parameter controls the AS-path attribute that is advertised with the aggregate route. If the switch has learnt multiple routes that are within the range of the aggregate address/mask, and the AS-paths associated with those routes contain different sets of AS-numbers, then it is not possible to create a single AS-path that accurately represents the AS-paths of all those component routes. In this case, the switch will, by default, advertise a NULL AS-path with the aggregate.

Usage [BGP4+] If the `summary-only` parameter is specified, then only the aggregate address/mask will be advertised, and none of the component addresses that fall within the range of the aggregate address/mask. For example, if you configure:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)#address-family ipv6
awplus(config-router-af)# aggregate-address 2001:0db8::/64
summary-only
```

then the switch will advertise the prefix `2001:0db8::/64`, but no component routes like `2001:0db8:010d::1/128`

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# aggregate-address 192.0.0.0/8 as-set
summary-only
```

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no aggregate-address 192.0.0.0/8 as-set
summary-only
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address family ipv6
awplus(config-router-af)# aggregate-address 2001:0db8::/64
as-set summary-only

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address family ipv6
awplus(config-router-af)# no aggregate-address 2001:0db8::/64
as-set summary-only

```

Related Commands [aggregate-address \(BGP and BGP4+\)](#) [match as-path \(Route Map\)](#)

auto-summary (BGP)

Use this command to enable sending summarized routes by a BGP speaker to its peers in the Router Configuration mode or in the Address-Family Configuration mode. BGP uses auto-summary to advertise summarized routes.

Use the **no** variant of this command to disable BGP auto-summary.

Syntax auto-summary
no auto-summary

Default The auto-summary function is disabled by default.

Mode Router Configuration and Address Family IPv4 mode

Usage If certain routes have already been advertised, enabling auto-summary results in non-summarized routes being withdrawn and only summarized routes are advertised. Summarized routes are advertised before non-summarized routes are withdrawn from all connected peers.

If certain routes have already been advertised, disabling auto-summary results in summarized routes being withdrawn and only non-summarized routes are advertised. Non-summarized routes are advertised before summarized routes are withdrawn from all connected peers.

Examples The following example enables auto-summary in Router Configuration mode:

```

awplus# configure
awplus(config)# router bgp 100
awplus(config-router)# auto-summary

```

The following example disables auto-summary in Router Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no auto-summary
```

The following example enables auto-summary in Address Family IPv4 mode:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv4
awplus(config-router-af)# auto-summary
```

The following example disables auto-summary in Address Family IPv4 mode:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no auto-summary
```

bgp aggregate-nexthop-check (BGP and BGP4+)

This command affects the operation of the summary-only option on the aggregate-address command.

This command enables a mode whereby the summary-only option will only suppress the component routes if those component routes all have the same nexthop. If the routes have different nexthops, then they will continue to be advertised to peers even if the summary-only option is configured. By default this is disabled.

The **no** variant of this command disables this function.

Syntax bgp aggregate-nexthop-check
no bgp aggregate-nexthop-check

Default Disabled by default.

Mode Global Configuration

Example

```
awplus# configure terminal
awplus(config)# bgp aggregate-nexthop-check
```

bgp always-compare-med (BGP and BGP4+)

This command enables BGP to compare the Multi Exit Discriminator (MED) for paths from neighbors in different autonomous systems.

Multi Exit Discriminator (MED) is used in best path selection by BGP. MED is compared after BGP attributes weight, local preference, AS-path and origin have been compared and are equal.

By default, MED comparison is done only among routes from the same autonomous system (AS). Use the **bgp always-compare-mode** command to allow comparison of MEDs from different ASs.

A path with a lower MED value is preferred. For example, if the bgp table contains the following entries, and the **bgp always-compare-med** command has been issued to enable this feature:

```
Route1: as-path 400, med 300
```

```
Route2: as-path 200, med 200
```

```
Route3: as-path 400, med 250
```

Route1 is compared to Route2. Route2 is best of the two (lower MED). Next, Route2 is compared to Route3 and Route2 is chosen best path again (lower MED). If **always-compare-med** was disabled, MED is not taken into account when Route1 and Route2 are compared, because of different ASs and MED is compared for only Route1 and Route3. In this case, Route3 would be the best path. The selected route is also affected by the **bgp deterministic-med** command. See the [bgp deterministic-med \(BGP and BGP4+\) command on page 3.29](#) for details.

If this command is used to compare MEDs for all paths, it should be configured on every BGP router in the AS.

The **no** variant of this command disallows the comparison.

Syntax `bgp always-compare-med`
`no bgp always-compare-med`

Default By default this feature is disabled.

Mode Router Configuration

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp always-compare-med
```

Related Commands [bgp bestpath med \(BGP and BGP4+\)](#)
[bgp bestpath as-path ignore \(BGP and BGP4+\)](#)
[bgp bestpath compare-routerid \(BGP and BGP4+\)](#)
[bgp deterministic-med \(BGP and BGP4+\)](#)

bgp bestpath as-path ignore (BGP and BGP4+)

This command prevents the router from considering as-path as a factor in the algorithm for choosing a route.

The **no** variant of this command allows the router to consider as-path in choosing a route.

Syntax `bgp bestpath as-path ignore`
`no bgp bestpath as-path ignore`

Mode Router Configuration

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath as-path ignore
```

Related Commands [bgp always-compare-med \(BGP and BGP4+\)](#)
[bgp bestpath med \(BGP and BGP4+\)](#)
[bgp bestpath compare-routerid \(BGP and BGP4+\)](#)

bgp bestpath compare-confed-aspath (BGP and BGP4+)

This command specifies that the AS confederation path length must be used, when available, in the BGP best path decision process. It is effective only when [bgp bestpath as-path ignore \(BGP and BGP4+\)](#) command has not been specified.

By default, if BGP receives routes with identical eBGP paths from eBGP peers, BGP does not continue to consider any AS confederation path length attributes that may be associated with the routes.

The **no** variant of this command returns the switch to the default state, where the switch ignores AS confederation path length in the BGP best path selection process.

Syntax `bgp bestpath compare-confed-aspath`
`no bgp bestpath compare-confed-aspath`

Mode Router Configuration

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath compare-confed-aspath
```

Related Commands [bgp bestpath as-path ignore \(BGP and BGP4+\)](#)

bgp bestpath compare-routerid (BGP and BGP4+)

By default, when comparing similar routes from peers, BGP does not consider the router ID of neighbors advertising the routes - BGP simply selects the first received route. Use this command to include router ID in the selection process; similar routes are compared and the route with the lowest router ID is selected.

The **no** variant of this command disables this feature, and returns the switch to the default state, where the switch ignores the router ID in the BGP best path selection process.

Syntax `bgp bestpath compare-routerid`
`no bgp bestpath compare-routerid`

Mode Router Configuration

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath compare-routerid
```

Related Commands [show ip bgp \(BGP\)](#)
[show bgp ipv6 neighbors \(BGP4+\)](#)

bgp bestpath med (BGP and BGP4+)

This command controls how the Multi Exit Discriminator (MED) attribute comparison is performed.

Use the **no** variant of this command to prevent BGP from considering the MED attribute when comparing paths.

Syntax `bgp bestpath med {[confed] [missing-as-worst]}`

Parameter	Description
<code>confed</code>	Compares MED among confederation paths.
<code>missing-as-worst</code>	Treats missing MED as the least preferred one.

Mode Router Configuration

Usage The **confed** parameter enables MED comparison among paths learned from confederation peers. The MED attributes are compared only if there is no external AS (Autonomous System), where an external AS is one that is not within the confederation. If there is an external AS in the path, then the MED comparison is not made.

For example, in the following paths the MED value is not compared with `Path3` since it is not in the confederation. MED is compared for `Path1` and `Path2` only.

`Path1 = 32000 32004, med=4`

`Path2 = 32001 32004, med=2`

`Path3 = 32003 1, med=1`

The effect of the **missing-as-worst** parameter is to treat a missing MED attribute in a path as having a value of infinity, making the path without a MED value the least desirable path. If the **missing-as-worst** parameter is not configured, the missing MED attribute is assigned the value of 0, making the path with the missing MED attribute the best path.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath med missing-as-worst

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath med confed

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath med confed missing-as-worst
```

Related Commands [bgp always-compare-med \(BGP and BGP4+\)](#)
[bgp bestpath as-path ignore \(BGP and BGP4+\)](#)
[bgp deterministic-med \(BGP and BGP4+\)](#)

bgp bestpath med remove-recv-med (BGP and BGP4+)

This command removes the Multi Exit Discriminator (MED) attribute from the update messages received by the BGP speaker from its peers. However, the local BGP speaker will send MED attributes in the update messages to its peers, unless specified not to by the **bgp bestpath med remove-send-med** command.

Use the **no** variant of this command to disable this feature.

Syntax `bgp bestpath med remove-recv-med`
`no bgp bestpath med remove-recv-med`

Mode Router Configuration

Example To enable the **remove-recv-med** feature on the BGP speaker belonging to the Autonomous System (AS) 100, enter the command:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath med remove-recv-med
```

Related Commands [bgp bestpath med remove-send-med \(BGP and BGP4+\)](#)

bgp bestpath med remove-send-med (BGP and BGP4+)

This command removes the Multi Exit Discriminator (MED) attribute from the update messages sent by the BGP speaker to its peers. However, the local BGP speaker will consider the MED attribute received from other peers during the decision and route selection process, unless specified not to by the **bgp bestpath med remove-recv-med** command.

Use the **no** variant of this command to disable this feature.

Syntax `bgp bestpath med remove-send-med`
`no bgp bestpath med remove-send-med`

Mode Router Configuration

Example To enable the **remove-send-med** feature on the BGP speaker belonging to the Autonomous System (AS) 100, enter the command:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp bestpath med remove-send-med
```

Related Commands [bgp bestpath med remove-recv-med \(BGP and BGP4+\)](#)

bgp client-to-client reflection (BGP and BGP4+)

This command restores route reflection from a BGP route reflector to clients, and is used to configure routers as route reflectors. Route reflectors are used when all Interior Border Gateway Protocol (iBGP) speakers are not fully meshed.

If the clients are fully meshed the route reflector is not required, use the **no** variant of this command to disable the client-to-client route reflection.

When a router is configured as a route reflector, client-to-client reflection is enabled by default.

The **no** variant of this command turns off client-to-client reflection.

Syntax `bgp client-to-client reflection`
`no bgp client-to-client reflection`

Default This command is enabled by default.

Mode Router Configuration

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp client-to-client reflection
```

Related Commands [bgp cluster-id \(BGP and BGP4+\)](#)
[neighbor route-reflector-client \(BGP\)](#)
[show bgp ipv6 \(BGP4+\)](#)
[show ip bgp \(BGP\)](#)

bgp cluster-id (BGP and BGP4+)

This command configures the cluster-id if the BGP cluster has more than one route reflector. A cluster includes one or more route reflectors and their clients. Usually, each cluster is identified by the router-id of its single route reflector. However, to increase redundancy, a cluster may sometimes have more than one route reflector. All router reflectors in such a cluster are then identified by a cluster-id.

The **bgp cluster-id** command is used to configure the 4 byte cluster ID for clusters with more than one route reflector.

The **no** variant of this command removes the cluster ID.

Syntax `bgp cluster-id {<ip-address>|<cluster-id>}`
`no bgp cluster-id`

Parameter	Description
<code><cluster-id></code>	<1-4294967295> Route Reflector cluster-id as a 32 bit quantity.
<code><ip-address></code>	A.B.C.D Route Reflector Cluster-id in IP address format.

Mode Router Configuration

Usage The following configuration creates `cluster-id 5` including two route-reflector-clients.

```
awplus(config)# router bgp 200
awplus(config-router)# neighbor 2.2.2.2 remote-as 200
awplus(config-router)# neighbor 3.3.3.3 remote-as 200
awplus(config-router)# neighbor 3.3.3.3 route-reflector-client
awplus(config-router)# neighbor 5.5.5.5 remote-as 200
awplus(config-router)# neighbor 5.5.5.5 route-reflector-client
awplus(config-router)# neighbor 6.6.6.6 remote-as 200
awplus(config-router)# bgp cluster-id 5
```

Examples To add a **bgp cluster-id**, apply the example commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp cluster-id 10.10.1.1
```

To remove a **bgp cluster-id** apply the example commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp cluster-id 10.10.1.1
```

Related Commands

- bgp client-to-client reflection (BGP and BGP4+)**
- neighbor route-reflector-client (BGP)**
- show bgp ipv6 (BGP4+)**
- show ip bgp (BGP)**

bgp confederation identifier (BGP and BGP4+)

This command specifies a BGP confederation identifier.

The **no** variant of this command removes all BGP confederation identifiers.

Syntax

```
bgp confederation identifier <1-4294967295>
no bgp confederation identifier
```

Parameter	Description
<1-4294967295>	Set routing domain confederation AS number.

Mode Router Configuration

Examples

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp confederation identifier 1

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp confederation identifier
```

Related Commands

- bgp confederation peers (BGP and BGP4+)**

bgp confederation peers (BGP and BGP4+)

This command configures the Autonomous Systems (AS) that belong to the same confederation as the current switch.

A confederation allows an AS to be divided into several sub-ASs. The overall AS is given a confederation identifier. External routers view only the whole confederation as one AS, whose AS number is the confederation identifier. Each sub-AS is fully meshed within itself and is visible internally to the confederation.

Use the **bgp confederation peer** command to define the list of AS numbers of the sub-ASs in the confederation containing the current switch.

The **no** variant of this command removes an autonomous system from the confederation.

Syntax `bgp confederation peers <1-4294967295>`
`no bgp confederation peers <1-4294967295>`

Parameter	Description
<code><1-4294967295></code>	AS numbers of eBGP peers that are under same confederation but in a different sub-AS.

Mode Router Configuration

Usage In the following configuration of **Router 1** the neighbor 172.210.30.2 and 172.210.20.1 have iBGP connection within AS 100. The neighbor 173.213.30.1 has an BGP connection, but it is within AS 200, which is part of the same confederation. The neighbor 6.6.6.6 has an eBGP connection to external AS 500.

In the configuration of **Router 2**, neighbor 5.5.5.4 has an eBGP connection to confederation 300. Router2 does not know about the ASs 100 and 200, it only knows about confederation 300.

Router 1

```
awplus(config)# router bgp 100
awplus(config-router)# bgp confederation identifier 300
awplus(config-router)# bgp confederation peers 200
awplus(config-router)# neighbor 172.210.30.2 remote-as 100
awplus(config-router)# neighbor 172.210.20.1 remote-as 100
awplus(config-router)# neighbor 173.213.30.1 remote-as 200
awplus(config-router)# neighbor 6.6.6.6 remote-as 300
```

Router 2

```
awplus(config)# router bgp 500
awplus(config-router)# neighbor 5.5.5.4 remote-as 300
```

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp confederation peers 1234
```

Related Commands [bgp confederation identifier \(BGP and BGP4+\)](#)

bgp config-type (BGP and BGP4+)

Use this command to set the BGP configuration type to either **standard** or **enhanced** types. When you configure the **enhanced** type, then BGP and BGP4+ communities are allowed to be sent and received by default. The **enhanced** type is configured by default.

Use the **no** variant of this command to restore the default BGP configuration type (**enhanced**).

Syntax `bgp config-type {standard|enhanced}`
`no bgp config-type`

Parameter	Description
standard	Specifies the industry standard style configuration. After setting the configuration to standard , make sure to use the neighbor send-community (BGP and BGP4+) command to send out BGP community attributes. The no synchronization command is enabled in the Global Configuration mode and is shown in the configuration.
enhanced	Specifies the enhanced style configuration. The enhanced configuration type requires no specific configuration for sending out BGP standard community and extended community attributes. The no synchronization command is enabled by default in the Global Configuration mode and is not shown in configuration output.

Default By default, the BGP configuration type is **enhanced**.

Mode Global Configuration

Usage Note that the **enhanced** type default configuration may cause issues in some networks if unauthorized BGP peers are advertising BGP communities to adjust routing decisions.

Changing modes requires you to **reload** your device for the change to take effect:

```
awplus(config)#bgp config-type standard
awplus(config)#exit
awplus#reload
reboot system? (y/n): y
```

When your device reloads, it will load with the standard BGP settings commonly used by most vendors. Apply the **standard** type configuration if you have interoperability issues.

Examples To specify the standard BGP configuration type, enter the following commands:

```
awplus# configure terminal
awplus(config)# bgp config-type standard
```

To specify the enhanced BGP configuration type, enter the following commands:

```
awplus# configure terminal
awplus(config)# bgp config-type enhanced
```

To restore the default BGP configuration type (enhanced), enter the following commands:

```
awplus# configure terminal
awplus(config)# no bgp config-type
```

Related Commands **neighbor send-community (BGP and BGP4+)**
synchronization (BGP and BGP4+)

bgp dampening (BGP and BGP4+)

This command enables BGP and BGP4+ dampening and sets BGP and BGP4+ dampening parameters. BGP4+ dampening is available from the IPv6 Address Family Configuration mode. BGP dampening is available from the Router Configuration mode.

The **no** variant of this command disables BGP dampening or unsets the BGP dampening parameters.

Syntax [BGP/BGP4+]

```

bgp dampening
no bgp dampening

bgp dampening <reachtime>
no bgp dampening <reachtime>

bgp dampening <reachtime> <reuse> <suppress> <maxsuppress>
<unreachtime>
no bgp dampening <reachtime> <reuse> <suppress> <maxsuppress>
<unreachtime>

bgp dampening route-map <routemap-name>
no bgp dampening route-map <routemap-name>

```

Parameter	Description
<reachtime>	<1-45> Specifies the reachability half-life time in minutes. The time for the penalty to decrease to one-half of its current value. The default is 15 minutes.
<reuse>	<1-20000> Specifies the reuse limit value. When the penalty for a suppressed route decays below the reuse value, the routes become unsuppressed. The default reuse limit is 750
<suppress>	<1-20000> Specifies the suppress limit value. When the penalty for a route exceeds the suppress value, the route is suppressed. The default suppress limit is 2000.
<maxsuppress>	<1-255> Specifies the max-suppress-time. Maximum time that a dampened route is suppressed. The default max-suppress value is 4 times the half-life time (60 minutes).
<unreachtime>	<1-45> Specifies the un-reachability half-life time for penalty, in minutes.
route-map	Route-map to specify criteria for dampening.
<routemap-name>	Specify the name of the route-map.

Mode [BGP] Router Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Route dampening minimizes the instability caused by route flapping. A penalty is added for every flap in a flapping route. As soon as the total penalty reaches the **suppress** limit the advertisement of the route is suppressed. This penalty is decayed according to the configured **half time** value. Once the penalty is lower than the **reuse** limit, the route advertisement is un-suppressed.

The dampening information is purged from the router once the penalty becomes less than half of the **reuse** limit.

**Example
[BGP]**

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# bgp dampening 20 800 2500 80 25
```

**Example
[BGP4+]**

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# address-family ipv6
awplus(config-router-af)# bgp dampening 20 800 2500 80 25
```

bgp default ipv4-unicast (BGP and BGP4+)

This command configures BGP defaults and activates IPv4-unicast for a peer by default. This affects BGP global configuration. By default, BGP exchanges IPv4 prefixes with a peer.

The **no** variant of this command disables this function. The BGP routing process will no longer exchange IPv4 addressing information with BGP neighbor routers. Note that disabling the exchange of IPv4 prefixes will also enable an IPv6 only BGP4+ network.

Syntax `bgp default ipv4-unicast`
`no bgp default ipv4-unicast`

Default This is enabled by default.

Mode Router Configuration

Usage Use the negated form of this command to enable an IPv6 only BGP4+ network.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp default ipv4-unicast
```

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp default ipv4-unicast
```

bgp default local-preference (BGP)

This command changes the default local preference value.

The local preference indicates the preferred path when there are multiple paths to the same destination. The path with the higher preference is preferred.

Use this command to define the default local preference value that the switch will advertise for the routes it sends. The preference is sent to all routers and access servers in the local autonomous system.

The **no** variant of this command reverts to the default local preference value of 100.

Syntax `bgp default local-preference <pref-value>`
`no bgp default local-preference [<pref-value>]`

Parameter	Description
<code><pref-value></code>	<code><0-4294967295></code> Configure default local preference value. The default local preference value is 100.

Default By default the local-preference value is 100.

Mode Router Configuration

Examples

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp default local-preference 2345555
```

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp default local-preference
```

bgp deterministic-med (BGP and BGP4+)

Use this command to allow or disallow the switch to compare the Multi Exit Discriminator (MED) variable when choosing among routes advertised by different peers in the same autonomous system (AS).

Use the **bgp deterministic-med** command to enable this feature to allow the comparison of MED variables when choosing among routes advertised by different peers in the same AS.

Use the **no** variant of this command to disable this feature to disallow the comparison of the MED variable when choosing among routes advertised by different peers in the same AS.

Syntax [BGP/BGP4+] `bgp deterministic-med`
`no bgp deterministic-med`

Default [BGP/BGP4+] Disabled

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] When the **bgp deterministic-med** command is enabled, routes from the same AS are grouped together and ordered according to their MED values, and the best routes of each group are compared.

The main benefit of this is that the choice of best route then does not depend on the order in which the routes happened to be received, which is rather random and arbitrary.

To see how this works, consider the following set of bgp table entries, all for the same route:

```
1: ASPATH 234, MED 120, internal, IGP metric to NEXT_HOP 40
2: ASPATH 389, MED 190, internal, IGP metric to NEXT_HOP 35
3: ASPATH 234, MED 245, external
```

If **bgp deterministic-med** is not enabled, then entry 3 : will be chosen, because it is an external route.

But if BGP deterministic-MED is enabled, the entries will be grouped as follows:

```
Group 1: 1: ASPATH 234, MED 120, internal, IGP metric to NEXT_HOP 40
          3: ASPATH 234, MED 245, external
```

```
Group 2: 2: ASPATH 389, MED 190, internal, IGP metric to NEXT_HOP 35
```

Note Routes from the same AS are grouped together and ordered by MED.



Entry 1 : is chosen as the best route from Group 1, since this route has the lowest MED value. Entry 2 has to be the best route in Group 2, since this is the only route in that group. These two group winners are compared against each other, and Entry 2 : is chosen as the best route because Entry 2 : has the lower metric to next-hop.

All routers in an AS should have the same setting for BGP deterministic-MED. All routers in an AS should have BGP deterministic-MED enabled with **bgp deterministic-med**, or all routers in an AS should have BGP deterministic-MED disabled with **no bgp-deterministic-med**.

In the example above, the MED values were not considered when comparing the winners of the two groups (the best routes from the different ASs). To use MED in the comparison of routes from different ASs, use the **bgp always-compare-med (BGP and BGP4+)** command on page 3.14.

Examples [BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp deterministic-med

awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp deterministic-med
```

Related Commands [show ip bgp \(BGP\)](#)
[show bgp ipv6 neighbors \(BGP4+\)](#)
[show ip bgp neighbors \(BGP\)](#)

bgp enforce-first-as (BGP and BGP4+)

Use this command to enforce the denying of eBGP updates in which the neighbor's AS number is not the first AS in the AS-path attribute.

Use the **no** variant of this command to disable this feature.

Syntax [BGP/BGP4+] `bgp enforce-first-as`
`no bgp enforce-first-as`

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] This command specifies that any updates received from an external neighbor that do not have the neighbor's configured Autonomous System (AS) at the beginning of the AS_PATH in the received update must be denied. Enabling this feature adds to the security of the BGP network by not allowing traffic from unauthorized systems.

Example [BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp enforce-first-as
```

bgp fast-external-failover (BGP and BGP4+)

Use this command to reset a BGP session immediately if the interface used for BGP connection goes down.

Use the **no** variant of this command to disable this feature.

Syntax [BGP/BGP4+] `bgp fast-external-failover`
`no bgp fast-external-failover`

Default [BGP/BGP4+] Enabled

Mode [BGP/BGP4+] Router Configuration

Example
[BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp fast-external-failover
```

bgp graceful-restart (BGP and BGP4+)

Use this command to enable BGP and BGP4+ graceful-restart capabilities for restart and stalepath times.

Use the **no** variant of this command to restore restart timers to their default settings.

Syntax `bgp graceful-restart`
 `[restart-time <delay-value>|stalepath-time <delay-value>]`
`no bgp graceful-restart [restart-time|stalepath-time]`

Parameter	Description
<code>restart-time</code>	The maximum time needed for neighbors to restart, in seconds. The default restart-time is 120 seconds.
<code>stalepath-time</code>	The maximum time to retain stale paths from restarting neighbors, in seconds. The default stalepath-time is 120 seconds.
<code><delay-value></code>	<1-3600> Maximum time in seconds. Default is 120 seconds.

Default The default BGP and BGP4+ graceful restart time is 120 seconds without restart-time or stalepath-time parameters specified. The default restart-time and stalepath-times are also 120 seconds.

Mode Router Configuration

Usage This command is used to set the maximum time that a graceful-restart neighbor waits to come back up after a restart. This value is applied to all neighbors unless you explicitly override it by configuring the corresponding value on the neighbor.

The **restart-time** parameter is used for setting the maximum time that a graceful-restart neighbor waits to come back up after a restart. This **restart-time** value is applied to neighbors unless you explicitly override it by configuring the corresponding value on the neighbor.

The **stalepath-time** parameter is used to set the maximum time to preserve stale paths from a gracefully restarted neighbor. All stalepaths, unless reinstated by the neighbor after a re-establishment, will be deleted when time, as specified by the **stalepath-time** parameter, expires.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# bgp graceful-restart 150

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no bgp graceful-restart
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# bgp graceful-restart restart-time 150
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no bgp graceful-restart restart-time 150
```

Related Commands [bgp graceful-restart graceful-reset \(BGP and BGP4+\)](#)
[restart bgp graceful \(BGP\)](#)

bgp graceful-restart graceful-reset (BGP and BGP4+)

This command enables BGP and BGP4+ graceful-restart when a configuration change forces a peer restart.

Use the **no** variant of this command to restore the device to its default state.

Syntax `bgp graceful-restart graceful-reset`
`no bgp graceful-restart graceful-reset`

Default Disabled

Mode Router Configuration

Usage The [bgp graceful-restart \(BGP and BGP4+\)](#) command on page 3.32 must be enabled before this command is enabled. All events that cause BGP peer reset, including all session reset commands, can trigger graceful-restart.

Example To enable the graceful-restart graceful-reset feature on the BGP or BGP4+ peer belonging to Autonomous System (AS) 10, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# bgp graceful-restart graceful-reset
```

To disable the graceful-restart graceful-reset feature on the BGP or BGP4+ peer belonging to Autonomous System (AS) 10, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no bgp graceful-restart graceful-reset
```

Related Commands [bgp graceful-restart \(BGP and BGP4+\)](#)

bgp log-neighbor-changes (BGP and BGP4+)

Use this command to enable logging of status change messages without turning on **debug bgp** commands.

Use the **no** variant of this command to disable this feature.

Syntax [BGP/BGP4+] `bgp log-neighbor-changes`
`no bgp log-neighbor-changes`

Default [BGP/BGP4+] Disabled

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] AlliedWare Plus™ provides other kinds of logging services for neighbor status, for example, **debug bgp fsm**, **debug bgp events**, etc.

However, these commands create a significant hit in the logging performance. If you need to log neighbor status changes only, we recommend turning off all the debug commands, and then use this command.

To see BGP neighbor changes in the log you must also set the log level to informational using the **log buffered** command.

A sample output of this log is:

```
%Protocol-Severity-Events: Message-text
```

A sample output of the log for an interface down event is:

```
%BGP-5-ADJCHANGE: neighbor 10.10.0.24 Down Interface flap
```

The **bgp log-neighbor-changes** command logs the following events:

- BGP Notification Received
- Erroneous BGP Update Received
- User reset request
- Peer time-out
- Peer Closing down the session
- Interface flap
- Router ID changed
- Neighbor deleted
- Member added to peer group
- Administrative shutdown
- Remote AS changed
- RR client configuration modification
- Soft reconfiguration modification

Example To enable the logging of BGP status changes without using the **debug bgp** command:
[BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp log-neighbor-changes
```

bgp memory maxallocation (BGP and BGP4+)

This command allocates a maximum percentage of the RAM (Random Access Memory) available on the switch for BGP processes.

When this percentage is exceeded, BGP peering terminates and an **out of resources** error displays. The default setting for **bgp memory maxallocation** is 100% memory allocation.

Use the **no** variant of this command to reset memory allocation to the default.

Syntax [BGP/BGP4+] `bgp memory maxallocation <1-100>`
`no bgp memory maxallocation`

Parameter	Description
<code><1-100></code>	Percentage of switch memory allocated to BGP processes. Note this is RAM (Random Access Memory), not switch flash memory.

Default [BGP/BGP4+] BGP processes are allocated the maximum percentage of 100% of the switch's available RAM memory by default. Note only non-default BGP memory allocation values are shown in the running or startup configuration files:

```
awplus#show running-config
!
bgp memory maxallocation 50
!
```

Mode [BGP/BGP4+] Global Configuration

Examples [BGP/BGP4+] To limit the maximum amount of memory used by BGP processes to 65% of the total RAM memory available on the switch, use the commands:

```
awplus# configure terminal
awplus(config)# bgp memory maxallocation 65
```

To return to the default 100% maximum RAM memory allocation available on the switch for BGP processes, use the commands:

```
awplus# configure terminal
awplus(config)# no bgp memory maxallocation
```

bgp nexthop-trigger-count (BGP and BGP4+)

Use this command to configure the display of BGP nexthop-tracking status.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] `bgp nexthop-trigger-count <0-127>`
`no bgp nexthop-trigger-count`

Parameter	Description
<code><0-127></code>	BGP nexthop-tracking status.

Mode [BGP/BGP4+] Router Configuration

Example [BGP/BGP4+] To enable nexthop-tracking status on the BGP peer belonging to the Autonomous System (AS) 100, enter the following commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp nexthop-trigger-count 10
```

To disable nexthop-tracking status, enter the following commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp nexthop-trigger-count
```

Related Commands [bgp nexthop-trigger delay \(BGP and BGP4+\)](#)
[bgp nexthop-trigger enable \(BGP and BGP4+\)](#)
[show bgp nexthop-tracking \(BGP\)](#)

bgp nexthop-trigger delay (BGP and BGP4+)

Use this command to set the delay interval for nexthop address tracking.

Use the **no** variant of this command to reset the timer value to the default.

Syntax [BGP/BGP4+] `bgp nexthop-trigger delay <1-100>`
`no bgp nexthop-trigger delay`

Parameter	Description
<1-100>	Nexthop trigger delay interval in seconds.

Default [BGP/BGP4+] The default nexthop delay interval is 5 seconds.

Mode [BGP/BGP4+] Global Configuration

Usage [BGP/BGP4+] This command configures the delay interval between routing table waits for nexthop delay tracking. The delay interval determines how long BGP waits after it receives the trigger from the system about one or more nexthop changes before it walks the full BGP table to determine which prefixes are affected by the nexthop changes.

Example To set the nexthop delay interval to 6 seconds, enter the command:

```
awplus# configure terminal
awplus(config)# bgp nexthop-trigger delay 6
```

Related Commands [bgp nexthop-trigger-count \(BGP and BGP4+\)](#)
[bgp nexthop-trigger enable \(BGP and BGP4+\)](#)

bgp nexthop-trigger enable (BGP and BGP4+)

Use this command to enable nexthop address tracking. If nexthop address tracking is enabled and a nexthop trigger delay interval has not been explicitly set with the **bgp nexthop-trigger delay (BGP and BGP4+)** command on page 3.38, the default delay interval of 5 seconds is used.

Use the **no** variant of this command to disable this feature.

Syntax [BGP/BGP4+] `bgp nexthop-trigger enable`
`no bgp nexthop-trigger enable`

Default [BGP/BGP4+] Disabled.

Mode [BGP/BGP4+] Global Configuration

Usage [BGP/BGP4+] Nexthop address tracking is an event driven notification system that monitors the status of routes installed in the Routing Information Base (RIB) and reports nexthop changes that affect internal BGP (iBGP) or external BGP (eBGP) prefixes directly to the BGP process. This improves the overall BGP convergence time, by allowing BGP to respond rapidly to nexthop changes for routes installed in the RIB.

If nexthop tracking is enabled after certain routes are learnt, the registration of all the nexthops of selected BGP routes are done immediately after the nexthop tracking feature is enabled.

If nexthop tracking is disabled, and if there are still some selected BGP routes, BGP deregisters the nexthops of all of the selected BGP routes from the system.

If nexthop tracking is disabled when nexthop tracking is in the process of execution, an error appears, and nexthop tracking is not disabled. However, if the nexthop tracking timer is running at the time of negation, the nexthop tracking timer is stopped, and nexthop tracking is disabled.

Example [BGP/BGP4+] To enable nexthop address tracking, enter the command:

```
awplus# configure terminal
awplus(config)# bgp nexthop-trigger enable
```

Related Commands **bgp nexthop-trigger-count (BGP and BGP4+)**
bgp nexthop-trigger delay (BGP and BGP4+)
show bgp nexthop-tracking (BGP)

bgp rfc1771-path-select (BGP)

Use this command to set the RFC1771 compatible path selection mechanism.

Use the **no** variant of this command to revert this setting.

Syntax `bgp rfc1771-path-select`
`no bgp rfc1771-path-select`

Default Industry standard compatible path selection mechanism.

Mode Global Configuration

Example

```
awplus# configure terminal
awplus(config)# bgp rfc1771-path-select
```

bgp rfc1771-strict (BGP)

Use this command to set the Strict RFC1771 setting.

Use the **no** variant of this command to revert this setting.

Syntax `bgp rfc1771-strict`
`no bgp rfc1771-strict`

Default Disabled

Mode Global Configuration

Example

```
awplus# configure terminal
awplus(config)# bgp rfc1771-strict
```

bgp router-id (BGP and BGP4+)

Use this command to configure the router identifier. The IPv4 address specified in this command does not have to be an IPv4 address that is configured on any of the interfaces on the switch. Note that you must specify an IPv4 address with this when used for BGP4+.

Use the **no** variant of this command to return the router-id to its default value (as described in **Default** below).

Syntax
[BGP/BGP4+] `bgp router-id <routerid>`
`no bgp router-id [<routerid>]`

Parameter	Description
<code><routerid></code>	Specify the IPv4 address without mask for a manually configured router ID, in the format A . B . C . D.

Default
[BGP/BGP4+] If the local loopback interface (lo) is configured the router-id is set to the IP address of this interface. If not, the highest IP address is the router-id.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage
[BGP/BGP4+] Use the **bgp router-id** command to manually configure a fixed router ID as a BGP or BGP4+ router identifier. The router-id configured using this command takes precedence over all other router-id sources

Examples
[BGP/BGP4+] To configure a router ID with an IPv4 address for a BGP or BGP4+ router identifier, enter the commands listed below:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp router-id 1.1.2.3
```

To disable the router ID for a BGP or BGP4+ router identifier enter the commands listed below:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no bgp router-id
```

bgp scan-time (BGP)

Use this command to set the interval for BGP route next-hop scanning.

Use the **no** variant of this command to disable this function.

Syntax `bgp scan-time <time>`
`no bgp scan-time [<time>]`

Parameter	Description
<time>	<0-60> Scanning interval in seconds.

Default The default scanning interval is 60 seconds.

Mode Router Configuration

Usage Use this command to configure scanning intervals of BGP routers. This interval is the period after which router checks the validity of the routes in its database.

To disable BGP scanning, set the scan time interval to 0 seconds.

Example

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# bgp scan-time 10
```

bgp update-delay (BGP and BGP4+)

Use this command to specify the update-delay value for a graceful-restart capable router.

Use the **no** variant of this command to revert to the default update-delay value.

Syntax [BGP/BGP4+] `bgp update-delay <1-3600>`
`no bgp update-delay [<1-3600>]`

Parameter	Description
<1-3600>	Delay value in seconds.

Default [BGP/BGP4+] The default update-delay value is 120 seconds.

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] The update-delay value is the maximum time a graceful-restart capable router which is restarting will defer route-selection and advertisements to all its graceful-restart capable neighbors. This maximum time starts from the instance the first neighbor attains established state after restart. The restarting router prematurely terminates this timer when end-of-rib markers are received from all its graceful-restart capable neighbors.

Example [BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# bgp update-delay 345
```

clear bgp * (BGP and BGP4+)

Use this command to reset the BGP and BGP4+ connections for all peers.

Syntax

```
clear bgp *
clear bgp * in [prefix-filter]
clear bgp * out
clear bgp * soft [in|out]
```

Parameter	Description
*	Clears all BGP and BGP4+ peers.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear bgp * soft in
awplus# clear bgp * in prefix-filter
```

clear bgp (IPv4 or IPv6 address) (BGP and BGP4+)

Use this command to reset the BGP and BGP4+ connections for specified peers. When VRF-Lite is configured, you can apply this command to a specific VRF instance. This command resets all BGP connections from any address family (from either IPv4 or IPv6 Address Families).

Syntax [BGP]

```
clear bgp <ip-addr>
clear bgp <ip-addr> in [prefix-filter]
clear bgp <ip-addr> out
clear bgp <ip-addr> soft [in|out]
```

Syntax [VRF]

```
clear ip bgp <ip-addr> [vrf <vrf-name>] [in|out|soft [in|out]]
```

Syntax [BGP4+]

```
clear bgp <ipv6-addr>
clear bgp <ipv6-addr> in [prefix-filter]
clear bgp <ipv6-addr> out
clear bgp <ipv6-addr> soft [in|out]
```

Parameter	Description
<ip-addr>	Specifies the IPv4 address of the neighbor whose connection is to be reset, entered in the form A.B.C.D.
<ipv6-addr>	Specifies the IPv6 address of the neighbor whose connection is to be reset, entered in hexadecimal in the format X:X::X:X.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.
vrf	Applies the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode [BGP/BGP4+] Privileged Exec

Examples [BGP]

```
awplus# clear bgp 3.3.3.3 soft in prefix-filter
awplus# clear bgp 2.2.2.2 out
```

To apply the above example to clear the BGP connection to peer at IP address 192.0.2.11 for the VRF instance blue, use the following commands:

```
awplus# clear bgp 192.0.2.11 vrf blue in
```

Examples [BGP4+]

```
awplus# clear bgp 2001:0db8:010d::1 soft in prefix-filter
awplus# clear bgp 2001:0db8:010d::1 out
```

Related Commands [clear bgp \(IPv4 or IPv6 address\) \(BGP and BGP4+\)](#)

clear bgp (ASN) (BGP and BGP4+)

Use this command to reset the BGP and BGP4+ connections for peers in the specified Autonomous System Number (ASN).

Syntax [BGP/BGP4+] `clear bgp <asn> [in [prefix-filter]|out|soft [in|out]]`

Parameter	Description
<asn>	<1-4294967295> The AS Number for which all routes will be cleared.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode [BGP/BGP4+] Privileged Exec

Examples [BGP/BGP4+]

```
awplus# clear bgp 300 soft in prefix-filter
awplus# clear bgp 500 soft out
awplus# clear bgp 300 soft in
awplus# clear bgp 1 in prefix-filter
```

clear bgp external (BGP and BGP4+)

Use this command to reset the BGP and BGP4+ connections for all external peers.

Syntax [BGP/BGP4+] `clear bgp external [in [prefix-filter]|out|soft [in|out]]`

Parameter	Description
external	Clears all external peers.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode [BGP/BGP4+] Privileged Exec

Examples [BGP/BGP4+]

```
awplus# clear bgp external soft in
```

```
awplus# clear bgp external in prefix-filter
```

clear bgp peer-group (BGP and BGP4+)

Use this command to reset the BGP and BGP4+ connections for all members of a peer group.

Syntax [BGP/BGP4+] `clear bgp peer-group <peer-group>
[in [prefix-filter]|out|soft [in|out]]`

Parameter	Description
peer-group	Clears all members of a peer group.
<peer-group>	Name of the BGP peer group
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode [BGP/BGP4+] Privileged Exec

Examples [BGP/BGP4+]

```
awplus# clear bgp peer-group P1 soft in
awplus# clear bgp peer-group P2 in
```

clear ip bgp * (BGP)

Use this command to reset all BGP connections, either by fully resetting sessions or by performing soft resets. If VRF-Lite is configured, you can reset BGP connections for all VRF instances or for a specified VRF instance.

Syntax

```
clear ip bgp *
clear ip bgp * in
clear ip bgp * out
clear ip bgp * soft [in|out]
clear ip bgp * in [prefix-filter]
```

Syntax [VRF]

```
clear ip bgp * [vrf <vrf-name>]
clear ip bgp * [vrf <vrf-name>] in
clear ip bgp * [vrf <vrf-name>] out
clear ip bgp * [vrf <vrf-name>] soft [in|out]
clear ip bgp * in [prefix-filter]
```

Parameter	Description
*	Clears all bgp peers.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.
vrf	Applies the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode Privileged Exec

Examples To clear all BGP peers, use the command:

```
awplus# clear ip bgp *
```

To clear all BGP peers in VRF instance red, use the command:

```
awplus# clear ip bgp * vrf red
```

To clear all outbound BGP peers in VRF instance red, use the command:

```
awplus# clear ip bgp * out vrf red
```

clear ip bgp (IPv4) (BGP)

Use this command to reset the IPv4 BGP connection to the peer specified by the IP address. When VRF-Lite is configured, you can apply this command to a specific VRF instance.

Syntax [BGP] `clear ip bgp <ipv4-addr> [in [prefix-filter]|out|soft [in|out]]`

Syntax [VRF] `clear ip bgp <ipv4-address> [vrf <vrf-name>] [in|out|soft [in|out]]`

Parameter	Description
<code><ipv4-addr></code>	Specifies the IPv4 address of the neighbor whose connection is to be reset, entered in the form A.B.C.D.
<code>in</code>	Indicates that incoming advertised routes will be cleared.
<code>prefix-filter</code>	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
<code>out</code>	Indicates that outgoing advertised routes will be cleared.
<code>soft in</code>	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
<code>soft out</code>	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.
<code>vrf</code>	Applies the command to the specified VRF instance.
<code><vrf-name></code>	The name of the VRF instance.

Mode [BGP] Privileged Exec

Examples [BGP] Use the following command to clear the BGP connection to peer at IPv4 address 192.168.1.1, and clearing all incoming routes.

```
awplus# clear ip bgp 192.168.1.1 in
```

To apply the above example to clear the BGP connection to peer at IP address 192.0.2.11 for the VRF instance blue, use the following commands:

```
awplus# clear ip bgp 192.0.2.11 vrf blue in
```

clear ip bgp dampening (BGP)

Use this command to clear route dampening information and unsuppress routes that have been suppressed.

Syntax `clear ip bgp dampening [<ip-address>|<ip-address/m>]`

Parameter	Description
<code><ip-address></code>	Specifies the IPv4 address for which BGP dampening is to be cleared, in dotted decimal format.
<code><ip-address/m></code>	Specifies the IPv4 address with mask for which BGP dampening is to be cleared, entered in the form A.B.C.D/M. Where M is the subnet mask
<code>ipv4</code>	Clears all IPv4 address family peers. Configure parameters relating to the BGP exchange of IPv4 prefixes.

Mode Privileged Exec

Examples

```
awplus# clear ip bgp dampening 10.10.0.121
```

clear ip bgp flap-statistics (BGP)

Use this command to clear the flap count and history duration for the specified prefixes.

Syntax `clear ip bgp flap-statistics [<ip-address>|<ip-address/m>]`

Parameter	Description
<code><ip-address></code>	Specifies the IPv4 address for which BGP flap count and history duration are to be cleared.
<code><ip-address/m></code>	Specifies the IPv4 address with mask for which BGP flap count and history duration are to be cleared.
<code>ipv4</code>	Clears all IPv4 address family peers. Configure parameters relating to the BGP exchange of IPv4 prefixes.

Mode Privileged Exec

Examples

```
awplus# clear ip bgp flap-statistics 10.10.0.121
```

clear ip bgp (ASN) (BGP)

Use this command to reset the BGP connections to all peers in a specified Autonomous System Number (ASN).

Syntax

```
clear ip bgp <asn> [in [prefix-filter]|out|soft [in|out]]
clear ip bgp <asn> ipv4
clear ip bgp <asn> ipv4 in [prefix-filter]
clear ip bgp <asn> ipv4 out
clear ip bgp <asn> ipv4 soft [in|out]
```

Parameter	Description
<asn>	<1-4294967295> Specifies the ASN for which all routes will be cleared.
ipv4	Clears all IPv4 address family peers. Configure parameters relating to the BGP exchange of IPv4 prefixes.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear ip bgp 100
```

clear ip bgp external (BGP)

Use this command to reset the BGP connections to all external peers.

Syntax `clear ip bgp external [in [prefix-filter]|out|soft [in|out]]`
`clear ip bgp external`
`clear ip bgp external in [prefix-filter]`
`clear ip bgp external out`
`clear ip bgp external soft [in|out]`

Parameter	Description
external	Clears all external peers.
ipv4	Clears all IPv4 address family peers. Configure parameters relating to the BGP exchange of IPv4 prefixes.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear ip bgp external out
```

clear ip bgp peer-group (BGP)

Use this command to reset the BGP connections to all members of a peer group.

Syntax

```
clear ip bgp peer-group <peer-name>
clear ip bgp peer-group <peer-name> in [prefix-filter]
clear ip bgp peer-group <peer-name> out
clear ip bgp peer-group <peer-name> soft [in|out]
clear ip bgp peer-group <peer-name> out
clear ip bgp peer-group <peer-name> soft [in|out]
```

Parameter	Description
peer-group	Clears all members of a peer group.
<peer-name>	Specifies the name of the peer group for which all members will be cleared.
ipv4	Clears all IPv4 address family peers. Configure parameters relating to the BGP exchange of IPv4 prefixes.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear ip bgp peer-group Peer1 out
```

clear bgp ipv6 (ipv6 address) (BGP4+)

Use this command to reset the IPv6 BGP4+ connection to the peer specified by the IP address.

Syntax `clear bgp ipv6 <ipv6-addr> [in [prefix-filter]|out|soft [in|out]]`

Parameter	Description
<code><ipv6-addr></code>	Specifies the IPv6 address of the neighbor whose connection is to be reset, entered in hexadecimal in the format X:X::X:X.
<code>ipv6</code>	Clears all IPv6 address family peers. Configure parameters relating to the BGP4+ exchange of IPv6 prefixes.
<code>in</code>	Indicates that incoming advertised routes will be cleared.
<code>prefix-filter</code>	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
<code>out</code>	Indicates that outgoing advertised routes will be cleared.
<code>soft in</code>	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
<code>soft out</code>	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples Use the following command to clear the BGP4+ connection to peer at IPv6 address 2001:0db8:010d::1, and clearing all incoming routes.

```
awplus# clear ip bgp 2001:0db8:010d::1 in
```

clear bgp ipv6 dampening (BGP4+)

Use this command to clear route dampening information and unsuppress routes that have been suppressed routes.

Syntax `clear bgp ipv6 dampening [<ipv6-addr>|<ipv6-addr/prefix-length>]`

Parameter	Description
<code><ipv6-addr></code>	Specifies the IPv6 address for which BGP4+ dampening is to be cleared, entered in hexadecimal in the format X:X::X:X.
<code><ipv6-addr/prefix-length></code>	Specifies the IPv6 address and prefix-length for which BGP4+ dampening is to be cleared. The IPv6 address uses the format X:X::X:X/Prefix-Length. The prefix-length is usually set between 0 and 64.

Mode Privileged Exec

Examples

```
awplus# clear bgp ipv6 dampening 2001:0db8:010d::1
awplus# clear bgp ipv6 dampening 2001:0db8::/64
```

clear bgp ipv6 flap-statistics (BGP4+)

Use this command to clear the flap count and history duration for the specified prefixes.

Syntax `clear bgp ipv6 flap-statistics
[<ipv6-addr>|<ipv6-addr/prefix-length>]`

Parameter	Description
<code><ipv6-addr></code>	Specifies the IPv6 address for which BGP4+ flap count and history duration are to be cleared, entered in hexadecimal in the format X:X::X:X.
<code><ipv6-addr/prefix-length></code>	Specifies the IPv6 address with with prefix length for which BGP4+ flap count and history duration are to be cleared. The IPv6 address uses the format X:X::X:X/Prefix-Length. The prefix-length is usually set between 0 and 64.

Mode Privileged Exec

Examples

```
awplus# clear bgp ipv6 flap-statistics 2001:0db8:010d::1
awplus# clear bgp ipv6 flap-statistics 2001:0db8::/64
```

clear bgp ipv6 (ASN) (BGP4+)

Use this command to reset the BGP4+ connections to all peers in a specified Autonomous System Number (ASN).

Syntax

```
clear bgp ipv6 <asn> [in [prefix-filter]|out|soft [in|out]]
clear bgp ipv6 <asn>
clear bgp ipv6 <asn> in [prefix-filter]
clear bgp ipv6 <asn> out
clear bgp ipv6 <asn> soft [in|out]
```

Parameter	Description
<asn>	<1-4294967295> Specifies the ASN for which all routes will be cleared.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear bgp ipv6 100
awplus# clear bgp ipv6 100 in
awplus# clear bgp ipv6 100 in prefix-filter
awplus# clear bgp ipv6 100 out
awplus# clear bgp ipv6 100 soft out
awplus# clear bgp ipv6 100 soft in
```

clear bgp ipv6 external (BGP4+)

Use this command to reset the BGP4+ connections to all external peers.

Syntax

```
clear bgp ipv6 external [in [prefix-filter]|out|soft [in|out]]
clear bgp ipv6 external
clear bgp ipv6 external in [prefix-filter]
clear bgp ipv6 external out
clear bgp ipv6 external soft [in|out]
```

Parameter	Description
external	Clears all external peers.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Examples

```
awplus# clear bgp ipv6 external in
awplus# clear bgp ipv6 external in prefix
awplus# clear bgp ipv6 external out
awplus# clear bgp ipv6 external soft out
awplus# clear bgp ipv6 external soft in
```

clear bgp ipv6 peer-group (BGP4+)

Use this command to reset the BGP4+ connections to all members of a peer group.

Syntax

```
clear bgp ipv6 peer-group <peer-name>
clear bgp ipv6 peer-group <peer-name> in [prefix-filter]
clear bgp ipv6 peer-group <peer-name> out
clear bgp ipv6 peer-group <peer-name> soft [in|out]
clear bgp ipv6 peer-group <peer-name>
clear bgp ipv6 peer-group <peer-name> in [prefix-filter]
clear bgp ipv6 peer-group <peer-name> out
clear bgp ipv6 peer-group <peer-name> soft [in|out]
```

Parameter	Description
peer-group	Clears all members of a peer group.
<peer-name>	Specifies the name of the peer group for which all members will be cleared.
ipv6	Clears all IPv6 address family peers. Configure parameters relating to the BGP4+ exchange of IPv6 prefixes.
in	Indicates that incoming advertised routes will be cleared.
prefix-filter	Specifies that a prefix-list will be sent, by the ORF mechanism, to those neighbors with which the ORF capability has been negotiated. The neighbors will be triggered to resend updates, which match the prefix-list filter, to the local router. The local router will then perform a soft reconfiguration.
out	Indicates that outgoing advertised routes will be cleared.
soft in	Soft inbound reset causes the neighbors to resend all their updates to the local switch, without resetting the connection or clearing the entries in the local switch. So, the local switch stores new updates, and uses them to systematically replace existing table entries. This process can use a considerable amount of memory.
soft out	Soft outbound reset causes the switch to simply resend all its updates to the specified neighbor(s), without resetting the connection, or clearing table entries.

Mode Privileged Exec

Example

```
awplus# clear bgp ipv6 peer-group Peer1 out
```

debug bgp (BGP)

Use this command to turn on one or more BGP debug options.

Use the **no** variant of this command to disable one or more BGP debug options.

Syntax

```
debug bgp [all|dampening|events|filters|fsm|keepalives|nht|nsm|
updates [in|out]]

no debug all bgp

no debug bgp [all|dampening|events|filters|fsm|keepalives|nht|nsm|
updates [in|out]]
```

Parameter	Description
all	Turns on all debugging for BGP.
dampening	Specifies debugging for BGP dampening.
events	Specifies debugging for BGP events.
filters	Specifies debugging for BGP filters.
fsm	Specifies debugging for BGP Finite State Machine (FSM).
keepalives	Specifies debugging for BGP keepalives.
nht	Specifies debugging for BGP NHT (Next Hop Tracking) messages.
nsm	Specifies debugging for NSM messages.
updates	[in out] Specifies debugging for BGP updates.
in	Inbound updates.
out	Outbound updates.

Mode Privileged Exec and Global Configuration

Usage If the command is entered with no parameters, then all debug options are enabled.

Examples

```
awplus# debug bgp
awplus# debug bgp events
awplus# debug bgp nht
awplus# debug bgp updates in
```

Related Commands [show debugging bgp \(BGP\)](#)
[undebug bgp \(BGP\)](#)

distance (BGP and BGP4+)

This command sets the administrative distance for BGP and BGP4+ routes. The switch uses this value to select between two or more routes to the same destination from two different routing protocols. Set the administrative distance for BGP routes in the Router Configuration mode, and for BGP4+ routes in IPv6 Address Family Configuration mode.

The route with the smallest administrative distance value is added to the Forwarding Information Base (FIB). For more information, see the **Administrative Distance** section in the **Route Selection** chapter of the **AlliedWare Plus Software Reference**.

The **no** variant of this command sets the administrative distance for the route to the default for the route type.

Syntax [BGP/BGP4+]

```
distance <1-255> <ip-address/m> [<listname>]
distance bgp <ebgp> <ibgp> <local>
no distance <1-255> <ip-address/m> [<listname>]
no distance bgp <ebgp> <ibgp> <local>
```

Parameter	Description
<1-255>	The administrative distance value you are setting for the route.
<ip-address/m>	The IP source prefix that you are changing the administrative distance for, entered in the form A.B.C.D/M. This is an IPv4 address in dotted decimal notation followed by a forward slash, and then the prefix length.
<listname>	The name of the access list to be applied to the administrative distance to selected routes.
<ebgp>	Specifies the administrative distance of external BGP (eBGP) routes. These are routes learnt from a neighbor out of the AS. Specify the distance as a number between 1 and 255. Default: 20
<ibgp>	Specifies the administrative distance of internal BGP (iBGP) routes. These are routes learnt from a neighbor within the same AS. Specify the distance as a number between 1 and 255. Default: 200
<local>	Specifies the administrative distance of local BGP routes. These are routes redistributed from another protocol within your device. Specify the distance as a number between 1 and 255. Default: 200

Mode [BGP] Router Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Usage You can use this command to set the administrative distance:

- for each BGP route types by specifying:
distance (BGP and BGP4+) <ebgp> <igbp> <local>
- for a specific route by specifying:
distance (BGP and BGP4+) <1-255> <ip-address/m> [<listname>]

If the administrative distance is changed, it could create inconsistency in the routing table and obstruct routing.

Examples [BGP] For BGP IPv4, to set the administrative distance to 34 for the route 10.10.0.0/24 in BGP 100, and use the access list "mylist" to filter the routes, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# distance 34 10.10.0.0/24 mylist
```

For BGP IPv4, to set BGP 100's administrative distances for eBGP routes to 34, iBGP routes to 23, and local BGP routes to 15, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# distance bgp 34 23 15
```

Example [BGP4+] For BGP4+ IPv6, to set BGP 100's administrative distances for eBGP routes to 34, iBGP routes to 23, and local BGP routes to 15, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# distance bgp 34 23 15
```

exit-address-family (BGP and BGP4+)

Use this command to exit either the IPv4 or the IPv6 Address Family Configuration mode.

Syntax [BGP/BGP4+] `exit-address-family`

Mode [BGP] IPv4 Address Family Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Examples [BGP] To enter and then exit IPv4 Address Family Configuration mode, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv4
awplus(config-router-af)# exit-address-family
awplus(config-router)#
```

To enter and then exit IPv4 Address Family Configuration mode for VRF instance red, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv4 vrf red
awplus(config-router-af)# exit-address-family
awplus(config-router)#
```

Example [BGP4+] To enter and then exit IPv6 Address Family Configuration mode, use the commands:

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# exit-address-family
awplus(config-router)#
```

Related Commands [address-family \(BGP and BGP4+\)](#)

ip as-path access-list (BGP and BGP4+)

This command defines a BGP and BGP4+ Autonomous System (AS) path access list.

The named AS path list is a filter based on regular expressions. If the regular expression matches the AS path in a BGP update message, then the permit or deny condition applies to that update. Use this command to define the BGP access list globally, then use neighbor configuration commands to apply the list to a particular neighbor.

The **no** variant of this command disables the use of the access list

Syntax [BGP/BGP4+] `ip as-path access-list <listname> {deny|permit} <reg-exp>`
`no ip as-path access-list <listname> {deny|permit} <reg-exp>`

Parameter	Description
<listname>	Specifies the name of the access list.
deny	Denies access to matching conditions.
permit	Permits access to matching conditions.
<reg-exp>	Specifies a regular expression to match the BGP AS paths.

Regular expressions listed below can be used with the **ip as-path-access-list** command:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

Mode [BGP/BGP4+] Global Configuration

Example [BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# ip as-path access-list mylist deny ^65535$
```

ip community-list (BGP and BGP4+)

Use this command to add an entry to a standard or extended BGP community-list filter.

Use the **no** variant of this command to delete a standard or extended community list entry.

Syntax [BGP/BGP4+]

```
ip community-list <listname> {deny|permit} .<community>
no ip community-list <listname> {deny|permit} .<community>
```

Parameter	Description
<listname>	Specifies the community listname.
deny	Specifies the community to reject.
permit	Specifies the community to accept.
.<community>	{<AS:VAL> local-AS no-advertise no-export }
<AS:VAL>	Specifies the valid value for the community number. This format represents the 32 bit communities value, where AS is the high order 16 bits and VAL is the low order 16 bits in digit format.
local-AS	Specifies routes not to be advertised to external BGP peers.
no-advertise	Specifies routes not to be advertised to other BGP peers.
no-export	Specifies routes not to be advertised outside of Autonomous System boundary.

Mode [BGP/BGP4+] Global Configuration

Usage [BGP/BGP4+] A community-list can be used as a filter to BGP updates. Use this command to define the community access list globally, then use neighbor configuration commands to apply the list to a particular neighbor.

There are two kinds of community-lists: expanded and standard. A standard community-list defines the community attributes explicitly and not via a regular expression. An expanded community-list defines the communities attributes with regular expressions.

The standard community-list is compiled into binary format and is directly compared with the BGP communities attribute in the BGP updates. The comparison is faster than the expanded community-list. Any community value that does not match the standard community value is automatically treated as expanded.

Example [BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# ip community-list mylist permit 7675:80 7675:90
```

Related Commands

- [ip community-list standard \(BGP and BGP4+\)](#)
- [ip community-list expanded \(BGP and BGP4+\)](#)
- [show ip community-list \(BGP and BGP4+\)](#)

ip community-list expanded (BGP and BGP4+)

Use this command to add an entry to an expanded BGP community-list filter.

Use the **no** variant of this command to delete the community list entry.

Syntax

```
ip community-list <100-199> {deny|permit} .<line>
no ip community-list <100-199> {deny|permit} .<line>
ip community-list expanded <expanded-listname> {deny|permit} .<line>
no ip community-list expanded <expanded-listname> {deny|permit}
.<line>
```

Parameter	Description
<100-199>	Expanded community list number.
expanded	Specifies an expanded community list.
<expanded-listname>	Expanded community list entry.
deny	Specifies community to reject.
permit	Specifies community to accept.
.<line>	Specifies community attributes with regular expressions.

Regular expressions listed below can be used with the **ip community-list expanded** command:

Symbo l	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

Mode Global Configuration

Usage A `community-list` can be used as a filter to BGP updates. Use this command to define the community access list globally, then use `neighbor` configuration commands to apply the list to a particular neighbor.

There are two kinds of community-lists: expanded and standard. A standard community-list defines the community attributes explicitly and not via a regular expression. An expanded community-list defines the communities attributes with regular expressions.

The standard community-list is compiled into binary format and is directly compared with the BGP communities attribute in the BGP updates. The comparison is faster than the expanded community-list. Any community value that does not match the standard community value is automatically treated as expanded.

Examples

```
awplus# configure terminal
awplus(config)# ip community-list 125 permit 6789906
awplus(config)# ip community-list expanded CLIST permit .*
```

Related Commands [ip community-list \(BGP and BGP4+\)](#)
[ip community-list standard \(BGP and BGP4+\)](#)
[show ip community-list \(BGP and BGP4+\)](#)

ip community-list standard (BGP and BGP4+)

Use this command to add an entry to a standard BGP community-list filter.

Use the **no** variant of this command to delete the standard community-list entry.

Syntax

```
ip community-list <1-99> {deny|permit} [.<community>]
no ip community-list <1-99> {deny|permit} [.<community>]
ip community-list standard <standard-listname> {deny|permit}
    [.<community>]
no ip community-list standard <standard-listname> {deny|permit}
    [.<community>]
```

Parameter	Description
<1-99>	Standard community list number.
standard	Specifies a standard community list.
<standard-listname>	Standard community list entry.
deny	Specifies community to reject.
permit	Specifies community to accept.
<community>	{<AS:VAL> local-AS no-advertise no-export }
<AS:VAL>	Specifies the valid value for the community number. This format represents the 32 bit communities value, where AS is the high order 16 bits and VAL is the low order 16 bits in digit format.
local-AS	Specifies routes not to be advertised to external BGP peers.
no-advertise	Specifies routes not to be advertised to other BGP peers.
no-export	Specifies routes not to be advertised outside of the Autonomous System boundary.

Mode Global Configuration

Usage A community-list can be used as a filter to BGP updates. Use this command to define the community access list globally, then use neighbor configuration commands to apply the list to a particular neighbor.

There are two kinds of community-lists: expanded and standard. The standard community-list defines the community attributes as explicit values, without regular expressions. The expanded community-list defines the communities attributes with regular expressions.

The standard community-list is compiled into binary format and is directly compared with the BGP communities attribute in the BGP updates. The comparison is faster than the expanded community-list. Any community value that does not match the standard community value is automatically treated as expanded.

Examples

```
awplus# configure terminal
awplus(config)# ip community-list standard CLIST permit 7675:80
7675:90 no-export
awplus(config)# ip community-list 34 permit 5675:50 no-
advertise
```

Related Commands [ip community-list \(BGP and BGP4+\)](#)
[ip community-list expanded \(BGP and BGP4+\)](#)
[show ip community-list \(BGP and BGP4+\)](#)

ip extcommunity-list expanded (BGP and BGP4+)

Use this command to create or delete an expanded extended community list.

Use the **no** variant of this command to delete the expanded extended community-list entry.

Syntax

```
ip extcommunity-list <100-199> {deny|permit}
    {.<line>|.<AS:NN>|.<ip-address>}

no ip extcommunity-list <100-199> {deny|permit}
    {.<line>|.<AS:NN>|.<ip-address>}

ip extcommunity-list expanded <expanded-listname> {deny|permit}
    {.<line>|.<AS:NN>|.<ip-address>}

no ip extcommunity-list expanded <expanded-listname> {deny|permit}
    {.<line>|.<AS:NN>|.<ip-address>}

no ip extcommunity-list <100-199>

no ip extcommunity-list expanded <expanded-listname>
```

Parameter	Description
<100-199>	Expanded extcommunity list number.
expanded	Specifies an expanded extcommunity list.
<expanded-listname>	Expanded extcommunity list entry.
deny	Specifies the extcommunity to reject.
permit	Specifies the extcommunity to accept.
.<line>	Specifies extcommunity attributes with regular expression.
<AS:NN>	Specifies the valid value for an extcommunity number. This format represents the 32 bit extcommunities value, where AA is the high order 16 bits and NN is the low order 16 bits in digit format.
<ip-address>	Specifies the IP address to deny or permit.

Regular expressions listed below are used with the **ip extcommunity-list expanded** command:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.

Symbol	Character	Meaning (cont'd)
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

Mode Global Configuration

Examples

```
awplus# configure terminal
awplus(config)# ip extcommunity-list 125 permit 4567335
awplus(config)# ip extcommunity-list expanded CLIST permit .*
```

Related Commands [ip extcommunity-list standard \(BGP and BGP4+\)](#)
[show ip extcommunity-list \(BGP and BGP4+\)](#)

ip extcommunity-list standard (BGP and BGP4+)

Use this command to create and delete a standard extended community list.

Use the **no** variant of this command to delete a standard extended community-list entry.

Syntax

```
ip extcommunity-list <1-99> {deny|permit} {rt|soo} <community-number>
ip extcommunity-list standard <standard-listname> {deny|permit}
  {rt|soo} <community-number>
no ip extcommunity-list <1-99> [{deny|permit} {rt|soo}
  <community-number>]
no ip extcommunity-list standard <standard-listname> [{deny|permit}
  {rt|soo} <community-number>]
```

Parameter	Description
<1-99>	Standard extcommunity list number.
standard	Specifies a standard extended community list.
<standard-listname>	Standard extended community list entry.
deny	Specifies the extended community to reject.
permit	Specifies the extended community to accept.
rt	Specifies the route target of the extended community.
soo	Specifies the site of origin of the extended community.
<community-number>	Specifies the valid value for an extended community number. This can be one of two formats: <ul style="list-style-type: none"> ■ <ASN:NN> where ASN is an AS (Autonomous System) number and NN is a value chosen by the ASN administrator ■ <A.B.C.D:NN> where A.B.C.D is an IPv4 address, and NN is a value chosen by the ASN administrator. Note that ASN and NN are both integers from 1 to 4294967295. AS numbers are assigned to the regional registries by IANA (www.iana.org) and must be obtained in your region.

Mode Global Configuration

Examples

```
awplus# configure terminal
awplus(config)# ip extcommunity-list 36 permit rt 5675:50
awplus(config)# ip extcommunity-list standard CLIST permit soo
7645:70
```

```
awplus# configure terminal
awplus(config)# ip extcommunity-list 36 deny rt 192.168.1.1:70
awplus(config)# ip extcommunity-list standard CLIST deny soo
10.10.1.1:50
```

Related Commands [ip extcommunity-list expanded \(BGP and BGP4+\)](#)
[show ip extcommunity-list \(BGP and BGP4+\)](#)

ip prefix-list (IPv4 Prefix List)

Use this command to create an entry for an IPv4 prefix list.

Use the **no** variant of this command to delete the IPv4 prefix-list entry.

Syntax

```
ip prefix-list <list-name> [seq <1-429496725>]
    {deny|permit}
    {any|<ip-prefix>}
    [ge <0-32>] [le <0-32>]

ip prefix-list <list-name> description <text>

ip prefix-list sequence-number

no ip prefix-list <list-name> [seq <1-429496725>]

no ip prefix-list <list-name> [description <text>]

no ip prefix-list sequence-number
```

Parameter	Description
<list-name>	Specifies the name of a prefix list.
seq <1-429496725>	Sequence number of the prefix list entry.
deny	Specifies that the prefixes are excluded from the list.
permit	Specifies that the prefixes are included in the list.
<ip-prefix>	Specifies the IPv4 address and length of the network mask in dotted decimal in the format A.B.C.D/M.
any	Any prefix match. Same as 0.0.0.0/0 le 32 .
ge <0-32>	Specifies the minimum prefix length to be matched.
le <0-32>	Specifies the maximum prefix length to be matched.
description <text>	Text description of the prefix list.
sequence-number	Specify sequence numbers included or excluded in prefix list.

Mode Global Configuration

Usage When the device processes a prefix list, it starts to match prefixes from the top of the prefix list, and stops whenever a permit or deny occurs. To promote efficiency, use the **seq** parameter and place common permits or denials towards the top of the list. If you do not use the **seq** parameter, the sequence values are generated in a sequence of 5.

The parameters **ge** and **le** specify the range of the prefix lengths to be matched. When setting these parameters, set the **le** value to be less than 32, and the **ge** value to be less than or equal to the **le** value and greater than the ip-prefix mask length.

Prefix lists implicitly exclude prefixes that are not explicitly permitted in the prefix list. This means if a prefix that is being checked against the prefix list reaches the end of the prefix list without matching a permit or deny, this prefix will be denied.

Example In the below sample configuration, the last `ip prefix-list` command in the below list matches all, and the first `ip prefix-list` command denies the IP network 76.2.2.0:

```

awplus(config)# router bgp 100
awplus(config-router)# network 172.1.1.0
awplus(config-router)# network 172.1.2.0
awplus(config-router)# neighbor 10.6.5.3 remote-as 300
awplus(config-router)# neighbor 10.6.5.3 prefix-list mylist out
awplus(config-router)# exit

awplus(config)# ip prefix-list mylist seq 5 deny
                        76.2.2.0/24

awplus(config)# ip prefix-list mylist seq 100 permit any
  
```

To deny the IP addresses between 10.0.0.0/14 (10.0.0.0 255.252.0.0) and 10.0.0.0/22 (10.0.0.0 255.255.252.0) within the 10.0.0.0/8 (10.0.0.0 255.0.0.0) addressing range, enter the following commands:

```

awplus# configure terminal
awplus(config)# ip prefix-list mylist seq 12345 deny 10.0.0.0/
                        8 ge 14 le 22
  
```

Related Commands [ipv6 prefix-list \(IPv6 Prefix List\)](#)
[show ip prefix-list \(IPv4 Prefix List\)](#)

ipv6 prefix-list (IPv6 Prefix List)

Use this command to create an IPv6 prefix list or an entry in an existing prefix list.

Use the **no** variant of this command to delete a whole prefix list or a prefix list entry.

Syntax

```

ipv6 prefix-list <list-name> [seq <1-429496725>]
    {deny|permit}
    {any|<ipv6-prefix>}
    [ge <0-128>] [le <0-128>]

ipv6 prefix-list <list-name> description <text>

no ipv6 prefix-list <list-name> [seq <1-429496725>]

no ipv6 prefix-list <list-name> [description <text>]

```

Parameter	Description
<list-name>	Specifies the name of a prefix list.
seq <1-429496725>	Sequence number of the prefix list entry.
deny	Specifies that the prefixes are excluded from the list.
permit	Specifies that the prefixes are included in the list.
<ipv6-prefix>	Specifies the IPv6 prefix and prefix length in hexadecimal in the format X:X::X:X/M.
any	Any prefix match. Same as ::0/0 le 128 .
ge <0-128>	Specifies the minimum prefix length to be matched.
le <0-128>	Specifies the maximum prefix length to be matched.
description	Prefix list specific description.
<text>	Up to 80 characters of text description of the prefix list.

Mode Global Configuration

Usage When the device processes a prefix list, it starts to match prefixes from the top of the prefix list, and stops whenever a permit or deny occurs. To promote efficiency, use the **seq** parameter and place common permits or denials towards the top of the list. If you do not use the **seq** parameter, the sequence values are generated in a sequence of 5.

The parameters **ge** and **le** specify the range of the prefix lengths to be matched. The parameters **ge** and **le** are only used if an ip-prefix is stated. When setting these parameters, set the **le** value to be less than 128, and the **ge** value to be less than or equal to the **le** value and greater than the ip-prefix mask length.

Prefix lists implicitly exclude prefixes that are not explicitly permitted in the prefix list. This means if a prefix that is being checked against the prefix list reaches the end of the prefix list without matching a permit or deny, this prefix will be denied.

Example To check the first 32 bits of the prefix 2001:db8:: and the subnet mask must be greater than or equal to 34 and less than or equal to 40, enter the following commands:

```
awplus# configure terminal
awplus(config)# ipv6 prefix-list mylist seq 12345 permit
                2001:db8::/32 ge 34 le 40
```

Related Commands [ip prefix-list \(IPv4 Prefix List\)](#)
[show ipv6 prefix-list \(IPv6 Prefix List\)](#)

match as-path (Route Map)

Use this command to add an autonomous system (AS) path match clause to a route map entry. Specify the AS path attribute value or values to match by specifying the name of an AS path access list. To create the AS path access list, enter Global Configuration mode and use the [ip as-path access-list \(BGP and BGP4+\)](#) command.

A BGP update message matches the route map if its attributes include AS path values that match the AS path access list.

Each entry of a route map can only match against one AS path access list in one AS path match clause. If the route map entry already has an AS path match clause, entering this command replaces that match clause with the new clause.

Note that AS path access lists and route map entries both specify an action of deny or permit. The action in the AS path access list determines whether the route map checks update messages for a given AS path value. The route map action and its **set** clauses determine what the route map does with update messages that contain that AS path value.

Use the **no** variant of this command to remove the AS path match clause from a route map entry.

Syntax `match as-path <as-path-listname>`
`no match as-path [<as-path-listname>]`

Parameter	Description
<code><as-path-listname></code>	Specifies an AS path access list name.

Mode Route-map Configuration

Usage This command is valid for BGP update messages only.

Example To add entry 34 to the route map called `myroute`, which will discard update messages if they contain the AS path values that are included in `myaccesslist`, use the commands:

```
awplus# configure terminal
awplus(config)# route-map myroute deny 34
awplus(config-route-map)# match as-path myaccesslist
```

Related Commands [ip as-path access-list \(BGP and BGP4+\)](#)
[route-map \(Route Map\)](#)

match community (Route Map)

Use this command to add a community match clause to a route map entry. Specify the community value or values to match by specifying a community list. To create the community list, enter Global Configuration mode and use the **ip community-list (BGP and BGP4+)** command.

A BGP update message matches the route map if its attributes include community values that match the community list.

Each entry of a route map can only match against one community list in one community match clause. If the route map entry already has a community match clause, entering this command replaces that match clause with the new clause.

Note that community lists and route map entries both specify an action of deny or permit. The action in the community list determines whether the route map checks update messages for a given community value. The route map action and its **set** clauses determine what the route map does with update messages that contain that community value.

Use the **no** variant of this command to remove the community match clause from a route map.

Syntax `match community {<community-listname> | <1-99> | <100-199>} [exact-match]`
`no match community [<community-listname> | <1-99> | <100-199>] [exact-match]`

Parameter	Description
<code><community-listname></code>	The community list name or number.
<code><1-99></code>	Community list number (standard range).
<code><100-199></code>	Community list number (expanded range).
<code>exact-match</code>	Exact matching of communities.

Mode Route-map Configuration

Usage This command is valid for BGP update messages only.

Communities are used to group and filter routes. They are designed to provide the ability to apply policies to large numbers of routes by using **match** and **set** commands. Community lists are used to identify and filter routes by their common attributes.

Example To add entry 3 to the route map called `myroute`, which will process update messages if they contain the community values that are included in `mylist`, use the commands:

```
awplus# configure terminal
awplus(config)# route-map myroute permit 3
awplus(config-route-map)# match community mylist
```

Related Commands [route-map \(Route Map\)](#)
[set community \(Route Map\)](#)

neighbor activate (BGP and BGP4+)

Use this command to enable the exchange of BGP IPv4 and BGP4+ IPv6 routes with a neighboring router, and also within either an IPv4 or an IPv6 specific address-family.

Use the **no** variant of this command to disable the exchange of information with a BGP or BGP4+ neighbor, in the Router Configuration or the Address Family Configuration mode.

Syntax [BGP/BGP4+] neighbor <neighborid> activate
no neighbor <neighborid> activate

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP] Use this command to enable the exchange of information to a neighbor. To exchange IPv4 or IPv6 prefixes with a BGP or a BGP4+ peer, you must configure this command for the peer or the peer group. This command only enables the exchange of information. You can establish peering without this command, but no prefixes and other information is sent until you apply this command to the neighbor.

This command triggers the switch to start a BGP or BGP4+ peering relationship with the specified BGP or BGP4+ neighbor and start exchanging routes with that neighbor. The command is required for neighbors configured in Address-Family Configuration mode, but it is not required in Router Configuration mode (that is, it does not affect the device's behavior).

Examples [BGP] To enable an exchange of routes with a neighboring router with the IPv4 address 10.10.10.1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 activate
```

To disable an exchange of routes with a neighboring router with the IPv4 address 10.10.10.1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 activate
```

To enable an exchange of routes in Address Family Configuration mode with a neighboring router with the IPv4 address 10.10.10.1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1 activate
```

To disable an exchange of routes in Address Family Configuration mode with a neighboring router with the IPv4 address 10.10.10.1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1 activate
```

To enable an exchange of routes with a neighboring router with the peer-group named group1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.63 remote-as 10
awplus(config-router)# neighbor 10.10.0.63 peer-group group1
awplus(config-router)# neighbor group1 activate
```

To disable an exchange of routes with a neighboring router with the peer-group named group1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 activate
```

Examples [BGP4+] To enable an exchange of routes in IPv6 Address Family Configuration mode with a neighboring router with the IPv6 address 2001:0db8:010d::1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 activate
```

To disable an exchange of routes in IPv6 Address Family Configuration mode with a neighboring router with the IPv6 address 2001:0db8:010d::1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
activate
```

To enable an exchange of routes with a neighboring router with the peer-group named group1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# neighbor group1 activate
```

To disable an exchange of routes with a neighboring router with the peer-group named group1, enter the commands as shown below:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 activate
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor advertisement-interval (BGP and BGP4+)

Use this command to set the minimum interval between sending iBGP or eBGP routing updates for a given route. This command reduces the flapping of individual routes.

Use the **no** variant of this command to set the interval time to the default values (30 seconds for eBGP peers and 5 seconds for iBGP peers) for a given route.

Syntax [BGP/BGP4+] `neighbor <neighborid> advertisement-interval <time>`
`no neighbor <neighborid> advertisement-interval [<time>]`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144 , and neighbor route-map (BGP and BGP4+) command on page 3.159 . When this parameter is used with this command, the command applies on all peers in the specified group. Note that if you apply an advertisement-interval value to a peer group it will apply to all members in the peer group.
<code><time></code>	<code><0-600></code> Advertisement -interval value in seconds.

Default [BGP/BGP4+] The default interval between sending routing updates for a given route to eBGP peers is 30 seconds, and the default interval for a given route to iBGP peers is 5 seconds.

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] Use this command to set the minimum interval between sending iBGP or eBGP routing updates for a given route. To reduce the flapping of routes to the internet, set a minimum advertisement interval, so iBGP or eBGP routing updates are sent per interval seconds.

BGP dampening can also be used to control the effects of flapping routes. See the [bgp dampening \(BGP and BGP4+\) command](#) in this chapter, and the [BGP and BGP4+ Route Dampening](#) section in the [BGP and BGP4+ Introduction](#) chapter for more information.

The advertisement-interval time value is the minimum time between the advertisement of Update messages sent from a BGP speaker to report changes to eBGP or iBGP peers. This is the minimum time between two Update messages sent to iBGP or eBGP peers.

See the [neighbor as-origination-interval \(BGP and BGP4+\) command](#) to set the interval time between messages to iBGP peers, which have prefixes within the local AS. Use this command instead of the [neighbor as-origination-interval \(BGP and BGP4+\) command](#) for eBGP peers with prefixes not in the same AS and updates not in a local AS.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.3
advertisement-interval 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.3
advertisement-interval
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.3 remote-as 10
awplus(config-router)# neighbor 10.10.0.3 peer-group group1
awplus(config-router)# neighbor group1 advertisement-interval
45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1
advertisement-interval
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
advertisement-interval 45

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
advertisement-interval

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# neighbor group1
advertisement-interval 45

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1
advertisement-interval

```

Validation Commands **show bgp ipv6 neighbors (BGP4+)**
show ip bgp neighbors (BGP)

Related Commands **neighbor as-origination-interval (BGP and BGP4+)**
neighbor peer-group (add a neighbor) (BGP and BGP4+)
neighbor route-map (BGP and BGP4+)

neighbor allows-in (BGP and BGP4+)

Use this command to accept an AS_PATH with the specified Autonomous System (AS) number from inbound updates for both BGP and BGP4+ routes.

This command allows BGP and BGP4+ to accept prefixes with the same ASN in the AS_PATH attribute. This command allows BGP and BGP4+ to accept up to 10 instances, configured by the *<occurrences>* placeholder, of its own AN in the AS_PATH for a prefix.

Use the **no** variant of this command to revert to default functionality (disabled by default).

Syntax [BGP/BGP4+] `neighbor <neighborid> allows-in <occurrences>`
`no neighbor <neighborid> allows-in`

Parameter	Description
<i><neighborid></i>	{ <i><ip-address></i> <i><ipv6-addr></i> <i><peer-group></i> }
<i><ip-address></i>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<i><ipv6-addr></i>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<i><peer-group></i>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<i><occurrences></i>	<i><1-10></i> Specifies the number of occurrences of the AS number.

Default [BGP/BGP4+] Disabled

Mode [BGP] Router Configuration or IPv4 Address Family Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use this command to configure PE (Provider Edge) routers to allow re-advertisement of all prefixes containing duplicate Autonomous System Numbers (ASNs). In a hub and spoke configuration, a PE router re-advertises all prefixes containing duplicate ASNs. Specify the remote-as or peer-group first using the related commands. The command allows a receiving peer to accept prefixes with its own AN in the AS_PATH, up the maximum number of instances, as configured by the *<occurrences>* placeholder.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.1 allows-in 3
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.0.1 allowas-in 3
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.1 allowas-in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.0.1 allowas-in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.1 remote-as 10
awplus(config-router)# neighbor 10.10.0.1 peer-group group1
awplus(config-router)# neighbor group1 allowas-in 3
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor group1 allowas-in 3
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
allowas-in 3
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
allowas-in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1
remote-as 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# neighbor group1 allowas-in 3
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor group1 allowas-in 3
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor as-origination-interval (BGP and BGP4+)

Use this command to adjust the sending of AS (Autonomous System) origination routing updates to a specified iBGP peer. This command adjusts the rate at which updates are sent to a specified iBGP peer (15 seconds by default). You must set a rate when you enable it.

The as-origination-interval is the minimum time set between the advertisement of Update messages sent from a BGP speaker to an iBGP peer to report changes within the local AS.

Use the **no** variant of this command to reset the timer to the default value of 15 seconds.

Syntax [BGP] `neighbor <neighbor_address> as-origination-interval <time>`
`no neighbor <neighbor_address> as-origination-interval [<time>]`

Syntax [BGP4+] `neighbor <ipv6-addr> as-origination-interval <time>`
`no neighbor <ipv6-addr> as-origination-interval [<time>]`

Parameter	Description
<code><neighbor_address></code>	Specify a neighbor IPv4 address, in dotted decimal in the format A.B.C.D.
<code><ipv6-addr></code>	Specify an address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><time></code>	<code><1-600></code> Time in seconds.

Default [BGP/BGP4+] The default interval between sending routing updates to iBGP peers, which include a prefix that originates from the local AS, is 15 seconds by default.

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] This command is used to change the minimum interval between sending AS-origination routing updates. The update interval for iBGP peers can be set from 1 to 600 seconds.

For interoperability with other vendors' devices, we recommend using the default value. The AS origination interval timer may not be available to adjust on other vendors' devices. Applying the default of 15 seconds across the AS maintains a common timer policy.

AlliedWare Plus devices use the default 15 second AS Origination Interval timer as per RFC 4271, a 30 second keepalive timer, a 90 second hold timer, a 120 second connect timer, a 5 second iBGP peer route advertisement interval, and a 30 second eBGP peer route advertisement interval.

Cisco devices use a 60 second keepalive timer, a 180 second hold timer, and no iBGP peer route interval timer (0). Juniper devices use a 10 second AS Origination Interval timer.

The as-origination-interval time value is the minimum amount of time between the advertisement of Update messages sent from a BGP speaker to report changes within the local AS. This is the minimum time between two Update messages to iBGP peers, which contain a prefix that originates from the same AS. See the [neighbor advertisement-interval \(BGP and BGP4+\)](#) command to set time between messages to eBGP peers.

Use this command instead of the [neighbor advertisement-interval \(BGP and BGP4+\)](#) command for iBGP peers with prefixes in the same AS for updates only within a local AS.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# neighbor 10.10.0.1
as-origination-interval 10
```

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no neighbor 10.10.0.1
as-origination-interval
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# neighbor 2001:0db8:010d::1
as-origination-interval 10
```

```
awplus# configure terminal
awplus(config)# router bgp 100
awplus(config-router)# no neighbor 2001:0db8:010d::1
as-origination-interval
```

Validation Commands **show bgp ipv6 neighbors (BGP4+)**
show ip bgp neighbors (BGP)

Related Commands **neighbor advertisement-interval (BGP and BGP4+)**
address-family (BGP and BGP4+)

neighbor attribute-unchanged (BGP and BGP4+)

Use this command to advertise unchanged BGP or BGP4+ attributes to the specified BGP or BGP4+ neighbor.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] `neighbor <neighborid> attribute-unchanged {as-path|next-hop|med}`
`no neighbor <neighborid> attribute-unchanged {as-path|next-hop|med}`

Parameter	Description
<neighborid>	{<ip-address> ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
as-path	AS path attribute.
next-hop	Next hop attribute.
med	Multi Exit Discriminator.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Note that specifying this command with the optional **as-path** parameter has the same effect as invoking the [neighbor transparent-as \(BGP and BGP4+\)](#) command.

Note this specifying this command with the optional **next-hop** parameter has the same effect as invoking the [neighbor transparent-nexthop \(BGP and BGP4+\)](#) command.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.75 attribute-unchanged
as-path med
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.75
attribute-unchanged as-path med

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.0.75
attribute-unchanged as-path med

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.0.75
attribute-unchanged as-path med

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.75 remote-as 10
awplus(config-router)# neighbor 10.10.0.75 peer-group group1
awplus(config-router)# neighbor group1 attribute-unchanged
as-path med

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 attribute-unchanged
as-path med
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
attribute-unchanged as-path med
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
attribute-unchanged as-path med
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# neighbor group1 attribute-unchanged
as-path med
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1
attribute-unchanged as-path med
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)
neighbor transparent-as (BGP and BGP4+)
neighbor transparent-nextthop (BGP and BGP4+)

neighbor capability graceful-restart (BGP and BGP4+)

Use this command to configure the device to advertise the Graceful Restart Capability to BGP and BGP4+ neighbors.

Use the **no** variant of this command to configure the device so it does not advertise the Graceful Restart Capability to its neighbor.

Syntax [BGP/BGP4+] `neighbor <neighborid> capability graceful-restart`
`no neighbor <neighborid> capability graceful-restart`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Default [BGP/BGP4+] Disabled

Mode [BGP] Router Configuration or IPv4 Address Family Configuration

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use the **neighbor capability graceful-restart** command to advertise to the BGP or BGP4+ neighbor routers the capability of graceful restart. First specify the BGP or BGP4+ neighbor's **remote-as** identification number as assigned by the neighbor router.

The graceful restart capability is advertised only when the graceful restart capability has been enabled using the **bgp graceful-restart (BGP and BGP4+)** command on page 3.32.

Examples [BGP]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.50 capability
graceful-restart

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.50 capability
graceful-restart

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.50 capability
graceful-restart

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.50 capability
graceful-restart

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.50 remote-as 10
awplus(config-router)# neighbor 10.10.10.50 peer-group group1
awplus(config-router)# neighbor group1 capability
graceful-restart

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 capability
graceful-restart

```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 capability
graceful-restart
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
capability graceful-restart
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# neighbor group1 capability
graceful-restart
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 capability
graceful-restart
```

Related Commands [bgp graceful-restart \(BGP and BGP4+\)](#)
[neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)
[restart bgp graceful \(BGP\)](#)

neighbor capability orf prefix-list (BGP and BGP4+)

Use this command to advertise ORF (Outbound Route Filters) capability to neighbors. Use this command to dynamically filter updates. The BGP speaker can advertise a prefix list with prefixes it wishes the peer to prune or filter from outgoing updates.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] `neighbor <neighborid> capability orf prefix-list {both|receive|send}`
`no neighbor <neighborid> capability orf prefix-list {both|receive|send}`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144 , and neighbor route-map (BGP and BGP4+) command on page 3.159 . When this parameter is used with this command, the command applies on all peers in the specified group.
<code>orf</code>	Advertises ORF capability to its neighbors.
<code>both</code>	Indicates that the local router can send ORF entries to its peer as well as receive ORF entries from its peer.
<code>receive</code>	Indicates that the local router is willing to receive ORF entries from its peer.
<code>send</code>	Indicates that the local router is willing to send ORF entries to its peer.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Default [BGP/BGP4+] Disabled

Usage [BGP/BGP4+] Outbound Route Filters (ORFs) send and receive capabilities to lessen the number of updates exchanged between neighbors. By filtering updates, this option minimizes generating and processing of updates. The local router advertises the ORF capability in **send** mode and the remote router receives the ORF capability in **receive** mode applying the filter as outbound policy. The two routers exchange updates to maintain the ORF for each router. Only an individual router or a peer-group can be configured to be in **receive** or **send** mode. A peer-group member cannot be configured in **receive** or **send** mode.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.5 capability orf
prefix-list both

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.5 capability orf
prefix-list both

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router)# neighbor 10.10.0.5 capability orf
prefix-list both

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router)# no neighbor 10.10.0.5 capability orf
prefix-list both

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.5 remote-as 10
awplus(config-router)# neighbor 10.10.0.5 peer-group group1
awplus(config-router)# neighbor group1 capability orf
prefix-list both

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 capability orf
prefix-list both
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router)# neighbor 2001:0db8:010d::1 capability
orf prefix-list both

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router)# no neighbor 2001:0db8:010d::1
capability orf prefix-list both

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# neighbor group1 capability orf
prefix-list both

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 capability orf
prefix-list both

```

Related Commands **neighbor capability orf prefix-list (BGP and BGP4+)**
neighbor peer-group (add a neighbor) (BGP and BGP+)
neighbor route-map (BGP and BGP4+)

neighbor capability route-refresh (BGP and BGP4+)

Use this command to advertise route-refresh capability to the specified BGP and BGP4+ neighbors.

Use the **no** variant of this command to disable this function

Syntax [BGP/BGP4+] `neighbor <neighborid> capability route-refresh`
`no neighbor <neighborid> capability route-refresh`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Default [BGP/BGP4+] Enabled

Usage [BGP/BGP4+] Use this command to advertise to peer about route refresh capability support. If route refresh capability is supported, then router can dynamically request that the peer readvertises its Adj-RIB-Out.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 capability route-
refresh
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 capability
route-refresh
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.1.1 remote-as 10
awplus(config-router)# neighbor 10.10.1.1 peer-group group1
awplus(config-router)# neighbor group1 capability route-refresh
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 capability route-
refresh
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 capability
route-refresh

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1 capability
route-refresh

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 capability
route-refresh

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 capability
route-refresh
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor collide-established (BGP and BGP4+)

Use this command to specify including a BGP or BGP4+ neighbor, already in an 'established' state, for conflict resolution when a TCP connection collision is detected.

Use the **no** variant of this command to remove a BGP or BGP4+ neighbor, already in an 'established' state, for conflict resolution when a TCP connection collision is detected.

Syntax [BGP/BGP4+] `neighbor <neighborid> collide-established`

`no neighbor <neighborid> collide-established`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] This command must be used only when specially required. It is not required in most network deployments.

The associated functionality of including an 'established' neighbor into TCP connection collision conflict resolution is automatically enabled when neighbor is configured for BGP graceful-restart.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 collide-established

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1
                           collide-established
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 collide-established

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 collide-established
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
collide-established

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
collide-established

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1
remote-as 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 collide-established

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 collide-established
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor default-originate (BGP and BGP4+)

Use this command to allow a BGP or BGP4+ local router to send the default route to a neighbor.

Use the **no** variant of this command to send no route as a default route.

Syntax [BGP/BGP4+] `neighbor {<neighborid>} default-originate [route-map <routemap-name>]`
`no neighbor {<neighborid>} default-originate [route-map <routemap-name>]`

Parameter	Description
<neighborid>	{ <ip-address> <ipv6-addr> <peer-group> }
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
route-map	If a route-map is specified, then the route table must contain at least one route that matches the permit criteria of the route map before the default route will be advertised to the specified neighbor.
<routemap-name>	Enter the route-map name.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 default-originate
route-map myroute
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1
                        default-originate route-map myroute
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1
                        default-originate route-map
                        myroute
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1
                        default-originate route-map myroute
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 default-originate
                        route-map myroute
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 default-originate
                        route-map myroute
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                        default-originate route-map myroute

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                        default-originate route-map myroute

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                        10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                        peer-group group1
awplus(config-router-af)# neighbor group1 default-originate
                        route-map myroute

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 default-originate
                        route-map myroute

```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor description (BGP and BGP4+)

Use this command to associate a description with a BGP or a BGP4+ neighbor. We recommend adding descriptions to defined neighbors, so any network administrators or network engineers can see a description of connected BGP or BGP4+ peers on the switch.

Use the **no** variant of this command to remove the description from a BGP or a BGP4+ neighbor.

Syntax `neighbor <neighborid> description <description>`
`no neighbor <neighborid> description [<description>]`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><description></code>	Enter up to 80 characters of text describing the neighbor.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 description Backup
router for sales

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 description
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 description Backup
router for sales
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 description Backup
router for sales.
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
description Backup router for sales
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
description
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 description Backup
router for sales
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 description Backup
router for sales

```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor disallow-infinite-holdtime (BGP and BGP4+)

Use this command to disallow the configuration of infinite holdtime for BGP and BGP4+.

Use the **no** variant of this command to allow the configuration of infinite holdtime for BGP or BGP4+.

Syntax [BGP] neighbor {<ip-address>} disallow-infinite-holdtime
no neighbor {<ip-address>} disallow-infinite-holdtime

Syntax [BGP4+] neighbor {<ipv6-addr>} disallow-infinite-holdtime
no neighbor {<ipv6-addr>} disallow-infinite-holdtime

Parameter	Description
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.

Mode [BGP/BGP4+] Router Configuration

Usage [BGP/BGP4+] This command enables the local BGP or BGP4+ speaker to reject holdtime "0" seconds from the peer during exchange of open messages or the user during configuration.

The **no** variant of this command allows the BGP speaker to accept "0" holdtime from the peer or during configuration.

Examples [BGP] To enable the **disallow-infinite-holdtime** feature on the BGP speaker with the IP address of 10.10.10.1, enter the command:

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 disallow-infinite-
holdtime

```

To disable the **disallow-infinite-holdtime** feature on the BGP speaker with the IP address of 10.10.10.10, enter the command:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1
disallow-infinite-holdtime
```

Examples [BGP4+] To enable the **disallow-infinite-holdtime** feature on the BGP4+ speaker with the IPv6 address of 2001:0db8:010d::1, enter the commands:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
disallow-infinite-holdtime
```

To disable the **disallow-infinite-holdtime** feature on the BGP4+ speaker with the IPv6 address of 2001:0db8:010d::1, enter the commands:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
disallow-infinite-holdtime
```

Related Commands [neighbor timers \(BGP and BGP4+\)](#)

neighbor distribute-list (BGP and BGP4+)

This command filters route updates from a particular BGP or BGP4+ neighbor using an access control list.

You can add one incoming and one outgoing distribute-list for each BGP or BGP4+ neighbor.

The **no** variant of this command removes a previously configured BGP or BGP4+ distribute-list.

Syntax [BGP/BGP4+] `neighbor <neighborid> distribute-list <access-list> {in|out}`
`no neighbor <neighborid> distribute-list <access-list> {in|out}`

Parameter	Description
<code><neighborid></code>	Specify an identification method for the BGP or BGP4+ peer. Use one of the following formats:
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation <i>A.B.C.D.</i>
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format <i>X:X::X:X.</i>
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) and neighbor route-map (BGP and BGP4+) commands. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><access-list></code>	The specific access-list used to filter routes. Specify one of the following types of access-lists:
<code><WORD></code>	The name of IP access-list.
<code><1-199></code>	The ID number of a standard IP access-list.
<code><1300-2699></code>	The ID number of an extended IP access-list.
<code>in</code>	Indicates that incoming advertised routes will be filtered.
<code>out</code>	Indicates that outgoing advertised routes will be filtered.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 distribute-list
mylist out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 distribute-list
mylist out

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1 distribute-list
mylist out

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1 distribute-
list mylist out

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor 10.10.10.1 distribute-list
mylist out

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 distribute-list
mylist out
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
distribute-list mylist out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                        distribute-list mylist out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                        10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
                        group1
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                        distribute-list mylist out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                        distribute-list mylist out
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor dont-capability-negotiate (BGP and BGP4+)

Use this command to disable capability negotiation for BGP and BGP4+.

The capability negotiation is performed by default. This command is used to allow compatibility with older BGP versions that have no capability parameters used in open messages between peers.

Use the **no** variant of this command to enable capability negotiation.

Syntax [BGP/BGP4+] `neighbor <neighborid> dont-capability-negotiate`
`no neighbor <neighborid> dont-capability-negotiate`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the IPv4 address of the BGP neighbor in dotted decimal, in the format <i>A . B . C . D</i> .
<code><ipv6-addr></code>	Specify the IPv6 address of the BGP4+ neighbor, entered in hexadecimal in the format <i>X:X::X:X</i> .
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) and neighbor route-map (BGP and BGP4+) commands. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.34
                        dont-capability-negotiate

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.34
                        dont-capability-negotiate
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.34 remote-as 100
awplus(config-router)# neighbor 10.10.10.34 peer-group group1
awplus(config-router)# neighbor group1
                        dont-capability-negotiate

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1
                        dont-capability-negotiate

```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 dont-
                        capability-negotiate

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
                        dont-capability-negotiate

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                        100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
                        group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1
                        dont-capability-negotiate

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1
                        dont-capability-negotiate
  
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor ebgp-multihop (BGP and BGP4+)

Use this command to accept and attempt BGP or BGP4+ connections to external peers on indirectly connected networks.

Effectively, this command sets the TTL value in the BGP or BGP4+ packets that the router sends to the neighbor, so that the packets may traverse the network route to the neighbor.

The switch will not establish a connection to a multihop neighbor, if the only route to the multihop peer is a default route.

Use the **no** variant of this command to return to the default.

Syntax [BGP/BGP4+] `neighbor <neighborid> ebgp-multihop [<count>]`
`no neighbor <neighborid> ebgp-multihop [<count>]`

Parameter	Description
<i><neighborid></i>	{ <i><ip-address></i> <i>ipv6-addr</i> <i><peer-group></i> }
<i><ip-addr></i>	Specify the address of an IPv4 BGP neighbor, entered in dotted decimal notation A.B.C.D.
<i><ipv6-addr></i>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<i><peer-group></i>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<i><count></i>	<i><1-255></i> The Maximum hop count, that is set in the TTL field of the BGP packets. If this optional parameter is not specified with the command, then the Maximum hop count is set to 255.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.34 remote-as 10
awplus(config-router)# neighbor 10.10.10.34 ebgp-multihop 5

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.34 ebgp-multihop 5

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.34 remote-as 10
awplus(config-router)# neighbor 10.10.10.34 peer-group group1
awplus(config-router)# neighbor group1 ebgp-multihop 5

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 ebgp-multihop 5
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as 10
awplus(config-router)# neighbor 2001:0db8:010d::1
awplus(config-router)# ebgp-multihop 5

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
awplus(config-router)# ebgp-multihop 5
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 ebgp-multihop 5

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 ebgp-multihop 5
```

Related Commands [neighbor ebgp-multihop \(BGP and BGP4+\)](#)
[neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor enforce-multihop (BGP and BGP4+)

Use this command to enforce the requirement that BGP and BGP4+ neighbors form multihop connections.

Use the **no** variant of this command to turn off this feature.

Syntax [BGP/BGP4+] neighbor <neighborid> enforce-multihop
no neighbor <neighborid> enforce-multihop

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	The address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.34 remote-as 10
awplus(config-router)# neighbor 10.10.0.34 enforce-multihop

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.34 enforce-multihop
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.34 remote-as 10
awplus(config-router)# neighbor 10.10.10.34 peer-group group1
awplus(config-router)# neighbor group1 enforce-multihop
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 enforce-multihop
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# neighbor 2001:0db8:010d::1
enforce-multihop
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
enforce-multihop
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 enforce-multihop
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 enforce-multihop
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor filter-list (BGP and BGP4+)

This command creates a BGP or BGP4+ filter using an AS (Autonomous System) path list. This command specifies an AS path list, which it then applies to filter updates to and from a BGP or a BGP4+ neighbor

The **no** variant of this command removes the previously specified BGP or BGP4+ filter using access control lists.

Syntax [BGP/BGP4+] `neighbor <neighborid> filter-list <listname> {in|out}`
`no neighbor <neighborid> filter-list <listname> {in|out}`

Parameter	Description
<code><neighborid></code>	Specify the identification method for the BGP or BGP4+ peer. Use one of the following formats:
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><listname></code>	Specify the name of an AS (Autonomous System) path list.
<code>in</code>	Indicates that incoming advertised routes will be filtered.
<code>out</code>	Indicates that outgoing advertised routes will be filtered.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] This command specifies a filter for updates based on a BGP AS (Autonomous System) path list.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.34 filter-list list1
out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.34 filter-list list1
out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.0.34 filter-list list1
out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.0.34 filter-list
list1 out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.34 remote-as 10
awplus(config-router)# neighbor 10.10.10.34 peer-group group1
awplus(config-router)# neighbor group1 filter-list list1 out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 filter-list list1 out
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
filter-list list1 out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
filter-list list1 out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# neighbor group1 filter-list list1 out
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 filter-list list1
out
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor interface (BGP4+)

Use this command to configure the interface name of a BGP4+ speaking neighbor.

Use the **no** variant of this command to disable this function.

Syntax [BGP4+] `neighbor {<ipv6-addr>|<ipaddress>} interface <interface>`
`no neighbor {<ipv6-addr>|<ipaddress>} interface <interface>`

Parameter	Description
<code><ipaddress></code>	Specifies the IPv4 address of the BGP neighbor - entered in dotted decimal notation in the format A.B.C.D.
<code><ipv6-addr></code>	Specifies the IPv6 address of the BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><interface></code>	Specifies the interface name of BGP neighbor, e.g. <code>vlan2</code> .

Mode [BGP4+] Router Configuration

Usage [BGP4+] This command is for use with BGP4+ peering. Use this command for BGP peering with IPv6 link local addresses.

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 interface vlan2

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 interface vlan2

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 interface
vlan2

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
interface vlan2
```

neighbor local-as (BGP and BGP4+)

Use this command to configure a local AS number for the specified BGP or BGP4+ neighbor. This overrides the local AS number specified by the [router bgp \(BGP and BGP4+\) command on page 3.200](#).

Use the **no** variant of this command to remove the local AS number for the specified BGP or BGP4+ neighbor.

Syntax [BGP/BGP4+] `neighbor <neighborid> local-as <as-number>`
`no neighbor <neighborid> local-as <as-number>`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	The address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) and neighbor route-map (BGP and BGP4+) commands. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><as-number></code>	<code><1-4294967295></code> Neighbor's Autonomous System (AS) number.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [VRF-Lite] When VRF-Lite is configured, this command allows internal BGP loopback connections between named VRFs and the default global routing instance to be configured to act as eBGP connections, instead of only iBGP.

Usage [BGP4+] When BGP4+ is configured, this command prepends the ASN as defined by the [router bgp \(BGP and BGP4+\)](#) command, and adds the ASN as defined by the [neighbor local-as \(BGP and BGP4+\)](#) command in front of the actual ASN as defined by the [router bgp \(BGP and BGP4+\)](#) command. This makes the peer believe it is peering with the ASN as defined by the [neighbor local-as \(BGP and BGP4+\)](#) command.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.34 local-as 1
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.34 local-as 1
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.34 remote-as 10
awplus(config-router)# neighbor 10.10.10.34 peer-group group1
awplus(config-router)# neighbor group1 local-as 1
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 local-as 1
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 local-as 1
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1 local-as 1
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 local-as 1
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 local-as 1
    
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)
[router bgp \(BGP and BGP4+\)](#)

neighbor maximum-prefix (BGP and BGP4+)

Use this command to control the number of prefixes that can be received from a BGP or a BGP4+ neighbor.

Use the **no** variant of this command to disable this function. Do not specify threshold to apply the default threshold of 75% for the maximum number of prefixes before this is applied.

Syntax [BGP/BGP4+] `neighbor <neighborid> maximum-prefix <maximum>`
`no neighbor <neighborid> maximum-prefix [<maximum>]`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><maximum></code>	<code><maxprefix></code> [<code><threshold></code>] [<code>warning-only</code>]
<code><maxprefix></code>	<code><1-4294967295></code> Specifies the maximum number of prefixes permitted.
<code><threshold></code>	<code><1-100></code> Specifies the threshold value, 1 to 100 percent. 75% by default.
<code>warning-only</code>	Only gives a warning message when the limit is exceeded.

Default [BGP/BGP4+] The default threshold value is 75%. If the threshold value is not specified this default is applied.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] The **neighbor maximum-prefix** command allows the configuration of a specified number of prefixes that a BGP or a BGP4+ router is allowed to receive from a neighbor. When the **warning-only** option is not used, if any extra prefixes are received, the router ends the peering. A terminated peer, stays down until the **clear ip bgp** command is used.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 maximum-prefix 1244
warning-only
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 maximum-prefix
1244 warning-only
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.72 remote-as 10
awplus(config-router)# neighbor 10.10.10.72 peer-group group1
awplus(config-router)# neighbor group1 maximum-prefix 1244
warning-only
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 maximum-prefix 1244
warning-only
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                           maximum-prefix 1244 warning-only
  
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                           maximum-prefix 1244 warning-only
  
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                           peer-group group1
awplus(config-router-af)# neighbor group1 maximum-prefix 1244
                           warning-only
  
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 maximum-prefix
                           1244 warning-only
  
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor next-hop-self (BGP and BGP4+)

Use this command to configure the BGP or BGP4+ router as the next hop for a BGP or BGP4+ speaking neighbor or peer group.

Use the **no** variant of this command to disable this feature.

Syntax [BGP/BGP4+] `neighbor <neighborid> next-hop-self`
`no neighbor <neighborid> next-hop-self`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] This command allows a BGP or BGP4+ router to change the nexthop information that is sent to the iBGP peer. The nexthop information is set to the IP address of the interface used to communicate with the neighbor.

This command can be run for a specific VRF instance.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 next-hop-self

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router)# neighbor 10.10.0.72 next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router)# no neighbor 10.10.0.72 next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.72 remote-as 10
awplus(config-router)# neighbor 10.10.10.72 peer-group group1
awplus(config-router)# neighbor group1 next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 next-hop-self
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# neighbor group1 next-hop-self
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 next-hop-self
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor override-capability (BGP and BGP4+)

Use this command to override a capability negotiation result for BGP and BGP4+.

Use the **no** variant of with this command to disable this function.

Syntax [BGP/BGP4+] `neighbor <neighborid> override-capability`

`no neighbor <neighborid> override-capability`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 override-capability

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72
                        override-capability
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.72 remote-as 10
awplus(config-router)# neighbor 10.10.10.72 peer-group group1
awplus(config-router)# neighbor group1 override-capability
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 override-capability
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor 2001:0db8:010d::1
override-capability
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor 2001:0db8:010d::1
override-capability
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 override-capability
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor group1 override-capability
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor passive (BGP and BGP4+)

Use this command to configure the local BGP or BGP4+ router to be passive with regard to the specified BGP or BGP4+ neighbor. This has the effect that the BGP or BGP4+ router will not attempt to initiate connections to this BGP or BGP4+ neighbor, but will accept incoming connection attempts from the BGP or BGP4+ neighbor.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] `neighbor <neighborid> passive`
`no neighbor <neighborid> passive`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 passive

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 passive
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.72 remote-as 10
awplus(config-router)# neighbor 10.10.10.72 peer-group group1
awplus(config-router)# neighbor group1 passive
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 passive
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 passive
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1 passive
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 passive
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 passive
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor password (BGP and BGP4+)

Use this command to enable MD5 authentication on a TCP connection between BGP and BGP4+ neighbors. No authentication is applied by default. To setup authentication for the session, you must first apply authentication on each connected peer for the session.

Use the **no** variant of this command to disable this function.

Syntax [BGP] `neighbor {<ip-address>|<peer-group-name>} password <password>`
`no neighbor {<ip-address>|<peer-group-name>} password [<password>]`

Syntax [BGP4+] `neighbor {<ipv6-addr>|<peer-group-name>} password <password>`
`no neighbor {<ipv6-addr>|<peer-group-name>} password [<password>]`

Parameter	Description
<code><ip-address></code>	Specifies the IP address of the BGP neighbor, in A.B.C.D format.
<code><ipv6-addr></code>	Specifies the IPv6 address of the BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group-name></code>	Name of an existing peer-group. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><password></code>	An alphanumeric string of characters to be used as password.

Default No authentication is applied by default.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] When using the `<peer-group-name>` parameter with this command (to apply this command to all peers in the group), see the related commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#) and [neighbor route-map \(BGP and BGP4+\)](#) for information about how to create peer groups first.

Examples [BGP] This example specifies the encryption type and the password (manager) for the neighbor 10.10.10.1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 password manager
```

This example removes the password set for the neighbor 10.10.10.1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 password
```

This example specifies the encryption type and the password (manager) for the neighbor peer group named group1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 password manager
```

This example removes the password set for the neighbor peer group named group1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 password
```

This example specifies the password (manager) for the neighbor peer group named group1 for an IPv4 address-family, VRF name red, and router bgp 10:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4 vrf red
awplus(config-router-af)# neighbor 10.10.10.1 password manager
```

This example removes the password (manager) for the neighbor peer group named group1 for an IPv4 address-family, VRF name red, and router bgp 10:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4 vrf red
awplus(config-router-af)# no neighbor 10.10.10.1 password
manager
```

This example specifies the password (manager) for the neighbor peer group named group1 for an IPv4 address-family, VRF name red, and router bgp 10:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group
group1
awplus(config-router)# address-family ipv4 vrf red
awplus(config-router-af)# neighbor group1 password manager
```

Examples [BGP4+] This example specifies the encryption type and the password (manager) for the neighbor 2001:0db8:010d::1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 password
manager
```

This example removes the password set for the neighbor 2001:0db8:010d::1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1 password
```

This example specifies the encryption type and the password (manager) for the neighbor peer group named group1:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 password manager
```

This example removes the password set for the neighbor peer group named `group1`:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 password
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor peer-group (add a neighbor) (BGP and BGP+)

Use this command to add a BGP or a BGP4+ neighbor to an existing peer-group.

Use the **no** variant of this command to disable this function.

Syntax [BGP] `neighbor <ip-address> peer-group <peer-group>`
`no neighbor <ip-address> peer-group <peer-group>`

Syntax [BGP4+] `neighbor <ipv6-addr> peer-group <peer-group>`
`no neighbor <ipv6-addr> peer-group <peer-group>`

Parameter	Description
<code><ip-address></code>	Specify the IPv4 address of the BGP neighbor, entered in the format A.B.C.D.
<code><ipv6-addr></code>	Specify the IPv6 address of the BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of the peer-group. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use this command to add neighbors with the same update policies to a peer group. This facilitates the updates of various policies, such as, distribute and filter lists. The peer-group is then configured easily with many of the neighbor commands. Any changes made to the peer group affect all members.

To create a peer-group use the [neighbor port \(BGP and BGP4+\)](#) command on page 3.147 and then use this command to add neighbors to the group.

Examples [BGP] This example shows a new peer-group `group1` and the addition of a neighbor `10.10.0.63` to the group.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.63 peer-group group1
```

This example shows a new peer-group `group1` and the removal of a neighbor `10.10.0.63` to the group.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# no neighbor 10.10.0.63 peer-group group1
```

Examples [BGP4+] This example shows a new peer-group `group1` and the addition of a neighbor `2001:0db8:010d::1` to the group.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
peer-group group1
```

This example shows a new peer-group `group1` and the removal of a neighbor `2001:0db8:010d::1` to the group.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
peer-group group1
```

Related Commands [neighbor peer-group \(create a peer-group\) \(BGP and BGP4+\)](#)
[neighbor port \(BGP and BGP4+\)](#)

neighbor peer-group (create a peer-group) (BGP and BGP4+)

Use this command to create a peer-group for BGP and BGP4+.

Use the **no** variant of this command to disable this function.

Syntax
[BGP/BGP4+]

```
neighbor <peer-group> peer-group
no neighbor <peer-group> peer-group
```

Parameter	Description
<peer-group>	Enter the name of the peer-group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage
[BGP/BGP4+]

Neighbors with the same update policies are grouped into peer groups. This facilitates the updates of various policies, such as, distribute and filter lists.

The peer-group is then configured easily with many of the neighbor commands. Any changes made to the peer group affect all members.

Use this command to create a peer-group, then use the **neighbor peer-group (add a neighbor) (BGP and BGP+)** command on page 3.144 to add neighbors to the group.

Examples
[BGP/BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 peer-group
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)

neighbor port (BGP and BGP4+)

Use this command to specify the TCP port to which packets are sent to on a BGP or a BGP4+ neighbor. TCP port 179 is the default port used to connect BGP and BGP4+ peers. You can specify a different destination port for the TCP session with this command.

Use the **no** variant of this command to reset the port number back to the default value (TCP port 179).

Syntax [BGP] `neighbor <neighborid> port <portnum>`
`no neighbor <neighborid> port [<portnum>]`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code>ipv6-addr</code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><portnum></code>	<code><0-65535></code> Specifies the TCP port number.

Default TCP port 179 is the default port used to connect BGP and BGP4+ peers.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor 10.10.10.10 port 643
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor 10.10.10.10 port 643
```

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 port 643

```

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor group1 port 643

```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor 2001:0db8:010d::1 port 643

```

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor 2001:0db8:010d::1 port 643

```

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(awplus-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 port 643

```

```

awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor group1 port 643

```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor prefix-list (BGP and BGP4+)

Use this command to distribute BGP and BGP4+ neighbor information as specified in a prefix list.

Use the **no** variant of this command to remove an entry.

Syntax [BGP/BGP4+] `neighbor <neighborid> prefix-list <listname> {in|out}`
`no neighbor <neighborid> prefix-list <listname> {in|out}`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><listname></code>	The name of an IP prefix list.
<code>in</code>	Specifies that the IP prefix list applies to incoming advertisements.
<code>out</code>	Specifies that the IP prefix list applies to outgoing advertisements.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use this command to specify a prefix list for filtering BGP or BGP4+ advertisements. Filtering by prefix list matches the prefixes of routes with those listed in the prefix list. If there is a match, the route is used. An empty prefix list permits all prefixes. If a given prefix does not match any entries of a prefix list, the route is denied access.

The router begins the search at the top of the prefix list, with the sequence number 1. Once a match or deny occurs, the router does not need to go through the rest of the prefix list. For efficiency the most common matches or denials are listed at the top.

The **neighbor distribute-list** command is an alternative to the **neighbor prefix-list** command and only one of them can be used for filtering to the same neighbor in any direction.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# ip prefix-list list1 deny 30.0.0.0/24
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 prefix-list list1 in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 prefix-list list1
in
```

```
awplus# configure terminal
awplus(config)# ip prefix-list list1 deny 30.0.0.0/24
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1 prefix-list list1
in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1 prefix-list
list1 in
```

```
awplus# configure terminal
awplus(config)# ip prefix-list list1 deny 30.0.0.0/24
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 prefix-list list1 in
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 prefix-list list1 in
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# ipv6 prefix-list list1 deny
                2001:0db8:010d::1/128

awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:: prefix-list
                           list1 in

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:: prefix-list
                           list1 in

awplus# configure terminal
awplus(config)# ip prefix-list list1 deny
                2001:0db8:010d::1/128

awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                           10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
                           group1
awplus(config-router-af)# neighbor group1 prefix-list list1 in

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 prefix-list list1
                           in
  
```

Related Commands

- [ip prefix-list \(IPv4 Prefix List\)](#)
- [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
- [neighbor route-map \(BGP and BGP4+\)](#)

neighbor remote-as (BGP and BGP4+)

Use this command to configure an internal or external BGP or BGP4+ (iBGP or eBGP) peering relationship with another router.

Use the **no** variant of this command to remove a previously configured BGP or BGP4+ peering relationship.

Syntax [BGP/BGP4+] neighbor <neighborid> remote-as <as-number>
no neighbor <neighborid> remote-as <as-number>

Syntax [VRF] neighbor <neighborid> remote-as <as-number> [global|vrf <vrf-name>]
no neighbor <neighborid> remote-as <as-number>

Parameter	Description
<neighborid>	{<ip-address> ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<as-number>	<1-4294967295> Neighbor's Autonomous System (AS) number.
global	Specify that the remote neighbor exists locally within the device, in the global routing domain
vrf	Specify that the remote neighbor exists locally within the device, in the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] This command is used to configure iBGP and eBGP peering relationships with other BGP or BGP4+ neighbors. A peer-group support of this command is configured only after creating a specific peer-group. Use the **no** variant of this command to remove a previously configured BGP peering relationship.

The **vrf** and **global** parameters are used to create internal 'loopback' BGP connections within the device between two VRF instances. This is used to leak BGP routes between a named VRF instance and the global routing instance. This requires BGP neighbors to be configured in both the global routing instance and in the named VRF instance.

Examples [BGP] To configure a BGP peering relationship from the neighbor with the IPv4 address 10.10.0.73 with another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.73 remote-as 10
```

To remove a configured BGP peering relationship from the neighbor with the IPv4 address 10.10.0.73 from another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.73 remote-as 10
```

To configure a BGP peering relationship from the neighbor with the peer group named group1 with another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 remote-as 10
```

To remove a configured BGP peering relationship from the neighbor with the peer group named group1 with another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 remote-as 10
```

Examples [BGP4+] To configure a BGP4+ peering relationship with another router:

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-
as 345
```

To remove a configured BGP4+ peering relationship from another router:

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# no neighbor 2001:0db8:010d::1
remote-as 345
```

To configure a BGP4+ peering relationship from the neighbor with the peer group named group1 with another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 remote-as 10
```

To remove a configured BGP4+ peering relationship from the neighbor with the peer group named group1 with another router:

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 remote-as 10
```

neighbor remove-private-AS (BGP)

Use this command to remove the private Autonomous System (AS) number from external outbound updates. Use the **no** variant of this command to revert to the default (disabled).

Syntax `neighbor <neighborid> remove-private-AS`
`no neighbor <neighborid> remove-private-AS`

Parameter	Description
<code><neighborid></code>	<code>{<ip-address> <tag>}</code>
<code><ip-address></code>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><tag></code>	Name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor remote-as (BGP and BGP4+) command on page 3.152. When this parameter is used with a command, the command applies on all peers in the specified group.

Default This command is disabled by default.

Mode BGP Router Configuration or BGP Router Address Family Configuration for a VRF instance

Usage The private AS numbers range from `<64512-65535>`. Private AS numbers are not advertised to the Internet. This command is used with external BGP peers only. The router removes the AS numbers only if the update includes private AS numbers. If the update includes both private and public AS numbers, the system treats it as an error.

This command removes private AS numbers for BGP in Router Configuration mode. This command is not supported for BGP4+ in IPv6 Address Family Configuration mode. This command removes a private AS number and makes an update packet with a public AS number as the AS path attribute. So only public AS numbers are entered in Internet BGP routing tables, and private AS numbers are not entered in Internet BGP tables.

For the filtering to apply, both peering devices must be set to use either 2-byte or extended 4- byte ASN (with the same ASN type set on both peers). For example, if a device (which defaults to use a 4-byte ASN), is peered with a device that defaults to a 2-byte ASN, then the device using a 2-byte ASN device also needs to be configured with the command **bgp extended-asn-cap** for the filtering to apply.

See the [How to remove private AS numbers for BGP](#) section in the [BGP and BGP4+ Introduction](#) chapter for further information about removing private AS numbers.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.63 remove-private-AS
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.63 remove-private-AS
```

Related Commands [show ip bgp \(BGP\)](#)

neighbor restart-time (BGP and BGP4+)

Use this command to set a different restart-time other than the global restart-time configured using the **bgp graceful-restart** command for BGP and BGP4+.

Use the **no** variant of this command to restore the device to its default state (see the default value of the **bgp graceful-restart** command).

Syntax [BGP/BGP4+]

```
neighbor <neighborid> restart-time <delay-value>
no neighbor <neighborid> restart-time <delay-value>
```

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<delay-value>	<1-3600> Delay value in seconds.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] This command takes precedence over the restart-time value specified using the **bgp graceful-restart** command.

The restart-time value is the maximum time that a graceful-restart neighbor waits to come back up after a restart. The default is 120 seconds.

Make sure that the restart time specified using this command does not exceed the stalepath-time specified in the Router Configuration mode.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 restart-time 45
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 restart-time 45
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 restart-time 45
```

Related Commands [bgp graceful-restart \(BGP and BGP4+\)](#)
[neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor route-map (BGP and BGP4+)

Use this command to apply a route map to incoming or outgoing routes for BGP or BGP4+.

Use the **no** variant of this command to remove a route map from a BGP or BGP4+ route.

Syntax [BGP/BGP4+] `neighbor <neighborid> route-map <mapname> {in|out}`
`no neighbor <neighborid> route-map <mapname> {in|out}`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<mapname>	Specifies name of the route-map.
in	Specifies that the access list applies to incoming advertisements.
out	Specifies that the access list applies to outgoing advertisements.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use the **neighbor route-map** command to filter updates and modify attributes. A route map is applied to inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates.

Examples [BGP] The following example shows the configuration of the route-map name **rmap2** and then the use of this map name in the **neighbor route-map** command for the neighbor with the IPv4 address 10.10.10.1 in the Router Configuration mode.

```
awplus# configure terminal
awplus(config)# route-map rmap2 permit 6
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 route-map rmap2
in
```

The following example shows the removal of the route-map name **rmap2** in the **neighbor route-map** command for the neighbor with the IPv4 address 10.10.10.1 in the Router Configuration mode.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 route-map
rmap2 in
```

The following example shows the configuration of the route-map name **rmap2** and then the use of this map name in the **neighbor route-map** command for the neighbor with the IPv4 address 10.10.10.1 in the IPv4 Address Family Configuration mode.

```
awplus# configure terminal
awplus(config)# route-map rmap2 permit 6
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1 route-map rmap2
in
```

The following example shows the removal of the route-map name **rmap2** in the **neighbor route-map** command for the neighbor with the IPv4 address 10.10.10.1 in the IPv4 Address Family Configuration mode.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1 route-map
rmap2 in
```

The following example shows the configuration of the route-map name **rmap2** and then the use of this map name in the **neighbor route-map** command for the neighbor with the peer group named **group1** in the Router Configuration mode.

```
awplus# configure terminal
awplus(config)# route-map rmap2 permit 6
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 route-map rmap2 in
```

The following example shows the removal the route-map name **rmap2** in the **neighbor route-map** command for the neighbor with the peer group named **group1** in the Router Configuration mode.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 route-map rmap2 in
```

Examples [BGP4+] The following example shows the configuration of the route-map name **rmap2** and then the use of this map name in the **neighbor route-map** command for the neighbor with the IPv6 address `2001:0db8:010d::1` in the IPv6 Address Family Configuration mode.

```
awplus# configure terminal
awplus(config)# route-map rmap2 permit 6
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 route-map
rmap2 in
```

The following example shows the removal of the route-map name **rmap2** in the **neighbor route-map** command for the neighbor with the IPv6 address `2001:0db8:010d::1` in the IPv6 Address Family Configuration mode.

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1 route-
map rmap2 in
```

The following example shows the configuration of the route-map name **rmap2** and then the use of this map name in the **neighbor route-map** command for the neighbor with the peer group named `group1` in the Router Configuration mode.

```

awplus# configure terminal
awplus(config)# route-map rmap2 permit 6
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# neighbor group1 route-map rmap2 in
  
```

The following example shows the removal the route-map name **rmap2** in the **neighbor route-map** command for the neighbor with the peer group named `group1` in the Router Configuration mode.

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 route-map rmap2 in
  
```

Related Commands

- [address-family \(BGP and BGP4+\)](#)
- [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
- [route-map \(Route Map\)](#)

neighbor route-reflector-client (BGP)

Use this command to configure the router as a BGP route reflector and configure the specified neighbor as its client.

Use the **no** variant of this command to indicate that the neighbor is not a client.

Syntax `neighbor <neighborid> route-reflector-client`
`no neighbor <neighborid> route-reflector-client`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><peer-group></code> }
<code><ip-address></code>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer-groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode Router Configuration or IPv4 Address Family Configuration for a VRF instance

Usage Route reflectors are a solution for the explosion of iBGP peering within an autonomous system. By route reflection the number of iBGP peers within an AS is reduced. Use the **neighbor route-reflector-client** command to configure the local router as the route reflector and specify neighbors as its client.

An AS can have more than one route reflector. One route reflector treats the other route reflector as another iBGP speaker.

In the following configuration, Router1 is the route reflector for clients 3.3.3.3 and 2.2.2.2; it also has a non-client peer 6.6.6.6:

```
Router1#
router bgp 200
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-reflector-client
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-reflector-client
neighbor 6.6.6.6 remote-as 200
```

Examples

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72
route-reflector-client
```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72
route-reflector-client
    
```

neighbor route-server-client (BGP)

Use this command to specify the peer as route server client.

Use the **no** variant of this command to disable this function.

Syntax neighbor <neighborid> route-server-client
 no neighbor <neighborid> route-server-client

Parameter	Description
<neighborid>	{<ip-address> <peer-group>}
<ip-address>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode Router Configuration

Examples

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 route-server-client

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72
route-server-client
    
```

neighbor send-community (BGP and BGP4+)

Use this command to specify that a community attribute should be sent to a BGP or BGP4+ neighbor.

Use the **no** variant of this command to remove the entry for the community attribute.

Syntax [BGP/BGP4+] `neighbor <neighborid> send-community {both|extended|standard}`
`no neighbor <neighborid> send-community {both|extended|standard}`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
both	Sends Standard and Extended Community attributes. Specifying this parameter with the no variant of this command results in no <code>standard</code> or <code>extended</code> community attributes being sent.
extended	Sends Extended Community attributes. Specifying this parameter with the no variant of this command results in no <code>extended</code> community attributes being sent.
standard	Sends Standard Community attributes. Specifying this parameter with the no variant of this command results in no <code>standard</code> community attributes being sent.

Default [BGP/BGP4+] Both **standard** and **extended** community attributes are sent to a neighbor.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration and IPv6 Address Family Configuration

Usage [BGP/BGP4+] This command is used to specify a community attribute to be sent to a neighbor. The community attribute groups destinations in a certain community and applies routing decisions according to those communities. On receiving community attributes the router reannounces them to the neighbor. Only when the **no** parameter is used with this command the community attributes are not reannounced to the neighbor.

By default, both **standard** and **extended** community attributes are sent to a neighbor.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 send-community
extended
```

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 send-community
extended
```

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.0.72 send-community
extended
```

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.0.72 send-community
extended
```

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 send-community extended
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
send-community extended

awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
send-community extended

awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
send-community extended

awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
send-community extended
```

```
awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 send-community
extended

awplus# configure terminal
awplus(config)# bgp config-type standard
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 send-community
extended
```

Related Commands **bgp config-type (BGP and BGP4+)**
neighbor peer-group (add a neighbor) (BGP and BGP+)
neighbor route-map (BGP and BGP4+)

neighbor shutdown (BGP and BGP4+)

Use this command to disable a peering relationship with a BGP or BGP4+ neighbor.

Use the **no** variant of this command to re-enable the BGP or BGP4+ neighbor.

Syntax [BGP/BGP4+] neighbor <neighborid> shutdown

no neighbor <neighborid> shutdown

Parameter	Description
<neighborid>	{<ip-address> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] This command shuts down any active session for the specified BGP or BGP4+ neighbor and clears all related routing data.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.72 shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.72 shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 shutdown
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1
shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 shutdown
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 shutdown
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor soft-reconfiguration inbound (BGP and BGP4+)

Use this command to configure the switch to start storing all updates from the BGP or BGP4+ neighbor, without any consideration of any inward route filtering policy that might be applied to the connection with this BGP or BGP4+ neighbor. This is so that the full set of the neighbor's updates are available locally to be used in a soft-reconfiguration event.

You may need to apply this older method of clearing routes if the peer does not support route refresh.

Use the **no** variant of this command to disable this function for a BGP or BGP4+ neighbor.

Syntax [BGP/BGP4+] `neighbor <neighborid> soft-reconfiguration inbound`
`no neighbor <neighborid> soft-reconfiguration inbound`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use this command to store updates for inbound soft reconfiguration. Soft-reconfiguration may be used in lieu of BGP route refresh capability. Using this command enables local storage of all the received routes and their attributes. This requires additional memory. When a soft reset (inbound) is done on this neighbor, the locally stored routes are re-processed according to the inbound policy. The BGP neighbor connection is not affected.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# neighbor 10.10.10.10
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor 10.10.10.10
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.10
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.10
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 soft-reconfiguration
                        inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# no neighbor group1 soft-reconfiguration
                        inbound
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                        soft-reconfiguration inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                        10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
                        group1
awplus(config-router-af)# neighbor group1 soft-reconfiguration
                        inbound
```

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1
                        soft-reconfiguration inbound
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor timers (BGP and BGP4+)

Use this command to set the keepalive, holdtime, and connect timers for a specific BGP or BGP4+ neighbor.

Use the **no** variant of this command to clear the timers for a specific BGP or BGP4+ neighbor.

Syntax [BGP/BGP4+] `neighbor <neighborid> timers {<keepalive> <holdtime>|connect <connect>}`

`no neighbor <neighborid> timers [connect]`

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP4+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<keepalive>	<0-65535> Frequency (in seconds) at which a router sends keepalive messages to its neighbor. The default is 30 seconds as per RFC 4271. The default Cisco IOS keepalive timer is 60 seconds.
<holdtime>	<0-65535> Interval (in seconds) after which, on not receiving a keepalive message, the router declares a neighbor dead. The default is 90 seconds as per RFC 4271. The default Cisco IOS keepalive timer is 180 seconds. Modify this value as needed for interoperability.
<connect>	connect <1-65535> Specifies the connect timer in seconds. The default connect timer value is 120 seconds as per RFC 4271. Modify this value as needed for interoperability.

Default The keepalive timer default is 60 seconds, the holdtime timer default is 90 seconds, and the connect timer default is 120 seconds as per RFC 4271. Holdtime is keepalive * 3.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] Keepalive messages are sent by a router to inform another router that the BGP connection between the two is still active. The keepalive interval is the period of time between each keepalive message sent by the router. The holdtime interval is the time the router waits to receive a keepalive message and if it does not receive a message for this period it declares the neighbor dead. The holdtime value must be 3 times the value of the keepalive value.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 timers 60 120
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 timers
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 timers 60 120
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 timers
```

Examples [BGP4+]

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1 timers 60
120

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 2001:0db8:010d::1 timers

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 timers 60 120

```

```

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 timers

```

Related Commands

- neighbor peer-group (add a neighbor) (BGP and BGP+)**
- neighbor route-map (BGP and BGP4+)**
- show ip bgp neighbors hold-time (BGP)**
- show ip bgp neighbors keepalive-interval (BGP)**
- timers (BGP and BGP4+)**

neighbor transparent-as (BGP and BGP4+)

Use this command to specify not to append your AS path number even if the BGP or BGP4+ peer is an eBGP peer.

Note this command has the same effect as invoking **neighbor attribute-unchanged (BGP and BGP4+)** and specifying the optional **as-path** parameter.

Syntax [BGP/BGP4+] neighbor <neighborid> transparent-as

Parameter	Description
<neighborid>	{ <ip-address> <ipv6-addr> <peer-group> }
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 transparent-as

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 transparent-as
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
transparent-as

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 transparent-as
```

Related Commands

- neighbor attribute-unchanged (BGP and BGP4+)**
- neighbor peer-group (add a neighbor) (BGP and BGP+)**
- neighbor route-map (BGP and BGP4+)**
- neighbor transparent-nextthop (BGP and BGP4+)**

neighbor transparent-nexthop (BGP and BGP4+)

Use this command to keep the nexthop value of the route even if the BGP or BGP4+ peer is an eBGP peer.

Note this command has the same effect as invoking **neighbor attribute-unchanged (BGP and BGP4+)** and specifying the optional **next-hop** parameter.

Syntax [BGP/BGP4+] neighbor <neighborid> transparent-nexthop

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.

Mode [BGP/BGP4+] Router Configuration

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 transparent-nexthop

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 transparent-nexthop
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 2001:0db8:010d::1
transparent-nexthop

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 transparent-nexthop
```

Related Commands [neighbor attribute-unchanged \(BGP and BGP4+\)](#)
[neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)
[neighbor transparent-as \(BGP and BGP4+\)](#)

neighbor unsuppress-map (BGP and BGP4+)

Use this command to selectively leak more specific routes to a particular BGP or BGP4+ neighbor.

Use the **no** variant of this command to remove selectively leaked specific routes to a particular BGP or BGP4+ neighbor.

Syntax [BGP/BGP4+] neighbor <neighborid> unsuppress-map <route-map-name>
no neighbor <neighborid> unsuppress-map <route-map-name>

Parameter	Description
<neighborid>	{<ip-address> <ipv6-addr> <peer-group>}
<ip-address>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<ipv6-addr>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<peer-group>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<route-map-name>	The name of the route-map used to select routes to be unsuppressed.

Mode [BGP] Router Configuration or Address-Family Configuration for IPv4 unicast or for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] When the [aggregate-address \(BGP and BGP4+\)](#) command is used with the **summary-only** option, the more-specific routes of the aggregate are suppressed to all neighbors. Use this command instead to selectively leak more-specific routes to a particular neighbor.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.0.73 unsuppress-map
mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4 unicast
awplus(config-router-af)# neighbor 10.10.0.70 unsuppress-map
mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.0.73 unsuppress-map
mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4 unicast
awplus(config-router-af)# no neighbor 10.10.0.70 unsuppress-map
mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 unsuppress-map mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 unsuppress-map
mymap
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6 unicast
awplus(config-router-af)# neighbor 2001:0db8:010d::1
                           unsuppress-map mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6 unicast
awplus(config-router-af)# no neighbor 2001:0db8:010d::1
                           unsuppress-map mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
                           10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
                           group1
awplus(config-router-af)# neighbor group1 unsuppress-map mymap

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 unsuppress-map
                           mymap
```

Related Commands [aggregate-address \(BGP and BGP4+\)](#)
[neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor update-source (BGP and BGP4+)

Use this command to specify the source IPv4 or IPv6 address of BGP or BGP4+ packets, which are sent to the neighbor for routing updates, as the IPv4 or IPv6 address configured on the specified interface. The specified interface is usually the local loopback (lo) interface to allow internal BGP or BGP4+ connections to stay up regardless of which interface is used to reach a neighbor.

Use the **no** variant of this command to remove the IPv4 or IPv6 address from the interface as the source IPv4 or IPv6 address of BGP or BGP4+ packets sent to the neighbor, and restores the interface assignment to the closest interface, which is also called the best local address.

Syntax [BGP/BGP4+] `neighbor <neighborid> update-source <interface>`
`no neighbor <neighborid> update-source`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><interface></code>	Specifies the local loopback interface (lo).

Default [BGP/BGP4+] Use of this command sets a default value of 2 for the maximum hop count.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration

Usage [BGP/BGP4+] Use this command in conjunction with any specified interface on the router. The local loopback interface is the interface that is most commonly used with this command. The use of local loopback interface eliminates a dependency and BGP or BGP4+ does not have to rely on the availability of a particular interface for making BGP or BGP4+ peer relationships.

Examples [BGP] To source BGP connections for neighbor 10.10.0.72 with the IP address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# interface lo
awplus(config-if)# ip address 10.10.0.73/24
awplus(config-if)# exit

awplus(config)# router bgp 100
awplus(config-router)# network 10.10.0.0
awplus(config-router)# neighbor 10.10.0.72 remote-as 110
awplus(config-router)# neighbor 10.10.0.72 update-source lo
```

To remove BGP connections for neighbor 10.10.0.72 with the IP address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# router bgp 100
awplus(config-router)# no neighbor 10.10.0.72 update-source
```

To source BGP connections for neighbor group1 with the IP address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# interface lo
awplus(config-if)# ip address 10.10.0.73/24
awplus(config-if)# exit

awplus(config)# router bgp 100
awplus(config-router)# network 10.10.0.0
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.0.72 remote-as 100
awplus(config-router)# neighbor 10.10.0.72 peer-group group1
awplus(config-router)# neighbor group1 update-source lo
```

To remove BGP connections for neighbor group1 with the IP address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# router bgp 100
awplus(config-router)# neighbor group1 update-source lo
```

Examples [BGP4+] To source BGP connections for neighbor 2001:0db8:010d::1 with the IPv6 address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# interface lo
awplus(config-if)# ipv6 address 2001:0db8:010d::1/128
awplus(config-if)# exit

awplus(config)# router bgp 100
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
110
awplus(config-router)# neighbor 2001:0db8:010d::1
update-source lo
```

To remove BGP connections for neighbor 2001:0db8:010d::1 with the IPv6 address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# router bgp 100
awplus(config-router)# no neighbor 2001:0db8:010d::1
update-source
```

To source BGP connections for neighbor group1 with the IPv6 address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# interface lo
awplus(config-if)# ipv6 address 2001:0db8:010d::1/128
awplus(config-if)# exit

awplus(config)# router bgp 100
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-group
group1
awplus(config-router-af)# exit
awplus(config-router)# neighbor group1 update-source lo
```

To remove BGP connections for neighbor group1 with the IPv6 address of the local loopback address instead of the best local address, enter the commands listed below:

```
awplus(config)# router bgp 100
awplus(config-router)# neighbor group1 update-source lo
```

Related Commands [neighbor peer-group \(add a neighbor\) \(BGP and BGP+\)](#)
[neighbor route-map \(BGP and BGP4+\)](#)

neighbor version (BGP)

Use this command to configure the switch to accept only a particular BGP version.

Use the **no** variant of this command to use the default BGP version (version 4).

Syntax `neighbor <neighborid> version <version>`
`no neighbor <neighborid> version`

Parameter	Description
<code><neighborid></code>	<code>{<ip-address> <peer-group>}</code>
<code><ip-address></code>	The address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP4+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><version></code>	{4} Specifies the BGP version number.

Mode Router Configuration or IPv4 Address Family Configuration for a VRF instance

Usage By default, the system uses BGP version 4 and on request dynamically negotiates down to version 2. Using this command disables the router's version-negotiation capability and forces the router to use only a specified version with the neighbor.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 version 4

awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 version 4
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 version
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 version
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

neighbor weight (BGP and BGP4+)

Use this command to set default weights for routes from this BGP or BGP4+ neighbor.

Use the **no** variant of this command to remove a weight assignment.

Syntax [BGP/BGP4+] `neighbor <neighborid> weight <weight>`
`no neighbor <neighborid> weight [<weight>]`

Parameter	Description
<code><neighborid></code>	{ <code><ip-address></code> <code><ipv6-addr></code> <code><peer-group></code> }
<code><ip-address></code>	Specify the address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
<code><ipv6-addr></code>	Specify the address of an IPv6 BGP4+ neighbor, entered in hexadecimal in the format X:X::X:X.
<code><peer-group></code>	Enter the name of an existing peer-group. For information on how to create peer groups, refer to the neighbor peer-group (add a neighbor) (BGP and BGP+) command on page 3.144, and neighbor route-map (BGP and BGP+) command on page 3.159. When this parameter is used with this command, the command applies on all peers in the specified group.
<code><weight></code>	<code><0-65535></code> Specifies the weight this command assigns to the route.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] Use this command to specify a weight value to all routes learned from a BGP or BGP4+ neighbor. The route with the highest weight gets preference when there are other routes on the network.

Unlike the local-preference attribute, the weight attribute is relevant only to the local router.

The weights assigned using the **set weight** command overrides the weights assigned using this command.

Examples [BGP]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor 10.10.10.1 weight 60
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor 10.10.10.1 weight
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# neighbor 10.10.10.1 weight 60
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv4
awplus(config-router-af)# no neighbor 10.10.10.1 weight
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 10.10.10.1 remote-as 10
awplus(config-router)# neighbor 10.10.10.1 peer-group group1
awplus(config-router)# neighbor group1 weight 60
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no neighbor group1 weight
```

Examples [BGP4+]

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 weight 60
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor 2001:0db8:010d::1 weight
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# neighbor group1 peer-group
awplus(config-router)# neighbor 2001:0db8:010d::1 remote-as
10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# neighbor 2001:0db8:010d::1 peer-
group group1
awplus(config-router-af)# neighbor group1 weight 60
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# address-family ipv6
awplus(config-router-af)# no neighbor group1 weight
```

Related Commands **neighbor peer-group (add a neighbor) (BGP and BGP+)**
neighbor route-map (BGP and BGP4+)

network (BGP and BGP4+)

Use this command to specify particular routes to be advertised into the BGP or BGP4+ routing process. A unicast network address without a mask is accepted if it falls into the natural boundary of its class. A class-boundary mask is derived if the address matches its natural class-boundary.

Note that you can specify a prefix length for the prefix being added, and you can also specify a classful network without a prefix length and an appropriate prefix length is added. Note that specifying a non-classful prefix without a prefix length results in a /32 prefix length on an IPv4 route.

Use the **no** variant of this command to remove a network route entry.

Syntax [BGP] `network {<ip-prefix/length>|<ip-network-addr>} [mask <network-mask>] [route-map <route-map-name>] [backdoor]`

`no network {<ip-prefix/length>|<ip-network-addr>} [mask <network-mask>] [route-map <route-map-name>] [backdoor]`

Syntax [BGP4+] `network {<ipv6-prefix/length>|<ipv6-network-addr>} [route-map <route-map-name>]`

`no network {<ipv6-prefix/length>|<ipv6-network-addr>} [route-map <route-map-name>]`

Parameter	Description
<code><ip-prefix/length></code>	IP network prefix and prefix length entered in dotted decimal format for the IP network prefix, then slash notation for the prefix length in the format A.B.C.D/M, e.g. 192.168.1.224/27
<code><ip-network-addr></code>	IP network prefix entered in dotted decimal format A.B.C.D, e.g. 192.168.1.224
<code><network-mask></code>	Specify a network mask in the format A.B.C.D, e.g. 255.255.255.224.
<code><ipv6-prefix/length></code>	IPv6 network prefix and prefix length entered in dotted decimal format for the IPv6 network prefix, then slash notation for the IPv6 prefix length in the format X::X::X/M, e.g. 2001:db8::/64
<code><ipv6-network-addr></code>	IP network prefix entered in dotted decimal format A.B.C.D, e.g. 192.168.1.224
<code><route-map-name></code>	Specify the name of the route map.
<code>backdoor</code>	Specify a BGP backdoor route that is not advertised.

Mode [BGP] Router Configuration and IPv4 Address Family [`ipv4 unicast`] mode

Mode [BGP4+] IPv6 Address Family Configuration

Usage [BGP/BGP4+] It does not matter how the route is arranged in the IP or IPv6 routing table. The route can arrive in the IP routing table by a static route, or the route can be learnt from OSPF or OSPFv3 or RIP or RIPng routing.

If you configure a route-map, then that route-map will be used in filtering the network, or the route-map will be used to modify the attributes that are advertised with the route.

Example [BGP] The following example illustrates a Class-A address configured as a network route. The natural Class-A network prefix mask length of 8 will be internally derived, that is, 2.0.0.0/8.

```
awplus(config)# router bgp 100
awplus(config-router)# network 2.0.0.0
```

Output [BGP] **Figure 3-1: Example output from the show running-config command after entering network 2.0.0.0**

```
awplus#show running-config
!
router bgp 100
 network 2.0.0.0/8
!
```

Example [BGP] The following example illustrates a network address which does not fall into its natural class boundary, and hence, is perceived as a host route, that is, 192.0.2.224/27.

```
awplus(config)# router bgp 100
awplus(config-router)# network 192.0.2.224 mask 255.255.255.224
```

Output [BGP] **Figure 3-2: Example output from the show running-config command after entering network 192.0.2.224 mask 255.255.255.224**

```
awplus#show running-config
!
router bgp 100
 network 192.0.2.224/27
!
```

Example [BGP] The following example is the same as the previous example for host route 192.0.2.224/27, but is entered in prefix/length format using slash notation (instead of prefix plus mask in dotted decimal format using the **mask** keyword before the network mask in dotted decimal format):

```
awplus(config)# router bgp 100
awplus(config-router)# network 192.0.2.224/27
```

Output [BGP] **Figure 3-3: Example output from the show running-config command after entering network 192.0.2.224/27**

```
awplus#show running-config
!
router bgp 100
  network 192.0.2.224/27
!
```

Example [BGP4+] The following example is the same as the previous example for host route 2001:db8::/32:

```
awplus(config)# router bgp 100
awplus(config-router)# address-family ipv6
awplus(config-router-af)# network 2001:db8::/32
```

Output [BGP4+] **Figure 3-4: Example output from the show running-config command after entering network 2001:db8::/32**

```
awplus#show running-config
!
router bgp 100
  network 2001:db8::/32
!
```

network synchronization (BGP and BGP4+)

Use this command to ensure the exact same static network prefix, specified through any of the **network** commands, is local or has IGP reachability before introduction to BGP or BGP4+.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] network synchronization
no network synchronization

Default [BGP/BGP4+] Network synchronization is disabled by default.

Mode [BGP] Router Configuration and IPv4 Address Family [ipv4 unicast] Configuration

Mode [BGP4+] IPv6 Address Family [ipv6 unicast] Configuration

Examples [BGP] The following example enables IGP synchronization of BGP static network routes in the Router Configuration mode.

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# network synchronization
```

The following example enables IGP synchronization of BGP static network routes in the IPv4-Unicast address family.

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# address-family ipv4 unicast
awplus(config-router-af)# network synchronization
```

Example [BGP4+] The following example enables IGP synchronization of BGP4+ static network routes in the IPv6-Unicast address family.

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# address-family ipv6 unicast
awplus(config-router-af)# network synchronization
```

redistribute (into BGP or BGP4+) (BGP and BGP4+)

Use this command to inject routes from one routing process into a BGP or BGP4+ routing table.

Use the **no** variant of this command to disable this function.

Syntax [BGP/BGP4+] `redistribute {ospf|rip|connected|static} [route-map <route-map-entry-pointer>]`
`no redistribute {ospf|rip|connected|static} [route-map <route-map-entry-pointer>]`

Parameter	Description
connected	Specifies the redistribution of connected routes for both BGP and BGP4+.
ospf	Specifies the redistribution of OSPF information for BGP or OSPFv3 information for BGP4+.
rip	Specifies the redistribution of RIP information for BGP or RIPng information for BGP4+.
static	Specifies the redistribution of Static routes for both BGP and BGP4+.
route-map	Route map reference for both BGP and BGP4+.
<route-map-entry-pointer>	Pointer to route-map entries.

Mode [BGP] Router Configuration or IPv4 Address Family Configuration for a VRF instance

Mode [BGP4+] Router Configuration or IPv6 Address Family Configuration

Usage [BGP/BGP4+] Redistribution is used by routing protocols to advertise routes that are learned by some other means, such as by another routing protocol or by static routes. Since all internal routes are dumped into BGP, careful filtering is applied to make sure that only routes to be advertised reach the internet, not everything. This command allows redistribution by injecting prefixes from one routing protocol into another routing protocol.

Examples [BGP/BGP+] The following example shows the configuration of a route-map named `rmap1`, which is then applied using the **redistribute route-map** command.

```
awplus# configure terminal
awplus(config)# route-map rmap1 permit 1
awplus(config-route-map)# match origin incomplete
awplus(config-route-map)# set metric 100
awplus(config-route-map)# exit
awplus(config)# router bgp 12
awplus(config-router)# redistribute ospf route-map rmap1
```

To apply the above example to a specific VRF instance named `blue`, use the following commands:

```
awplus(config)# router bgp 12
awplus(config-router)# address-family ipv4 vrf blue
awplus(config-router-af)# redistribute ospf route-map rmap1
```

The following example shows the configuration of a route-map named `rmap2`, which is then applied using the **redistribute route-map** command.

```
awplus# configure terminal
awplus(config)# route-map rmap2 permit 3
awplus(config-route-map)# match interface vlan1
awplus(config-route-map)# set metric-type 1
awplus(config-route-map)# exit
awplus(config)# router ospf 100
awplus(config-router)# redistribute bgp route-map rmap2
```

Note that configuring a route-map and applying it with the `redistribute route-map` command allows you to filter which routes are distributed from another routing protocol (such as OSPF with BGP). A route-map can also set the metric, tag, and metric-type of the redistributed routes.

restart bgp graceful (BGP)

Use this command to force the switch to perform a graceful BGP restart.

Syntax `restart bgp graceful`

Mode Privileged Exec

Usage Before using this command, BGP graceful-restart capabilities must be enabled within the router BGP ([bgp graceful-restart \(BGP and BGP4+\) command on page 3.32](#)), and each neighbor configured on the device should be set to advertise its graceful-restart capability ([bgp graceful-restart graceful-reset \(BGP and BGP4+\) command on page 3.33](#)). The neighbor devices also need to have BGP graceful-restart capabilities enabled ([bgp graceful-restart \(BGP and BGP4+\) command on page 3.32](#)).

This command stops the whole BGP process and makes the switch retain the BGP routes and mark them as stale. Receiving BGP speakers, retain and mark as stale all BGP routes received from the restarting speaker for all the address families received in the Graceful Restart Capability exchange.

When a **restart bgp graceful** command is issued, the BGP configuration is reloaded from the last saved configuration. Ensure you first issue a **copy running-config startup-config**.

Example

```
awplus# restart bgp graceful
```

Related Commands [bgp graceful-restart \(BGP and BGP4+\)](#)
[bgp graceful-restart graceful-reset \(BGP and BGP4+\)](#)

router bgp (BGP and BGP4+)

Use this command to configure a BGP routing process, specifying the 32-bit Autonomous System (AS) number.

Use the **no** variant of this command to disable a BGP routing process, specifying the 32-bit AS number.

Syntax `router bgp <asn>`
`no router bgp <asn>`

Parameter	Description
<asn>	<1-4294967295> Specifies the 32-bit Autonomous System (AS) number.

Mode Global Configuration

Usage The **router bgp** command enables a BGP routing process:

```
router bgp 1
  neighbor 10.0.0.1 remote-as 1
  neighbor 10.0.0.2 remote-as 1
  !
router bgp 2
  neighbor 10.0.0.3 remote-as 2
  neighbor 10.0.0.4 remote-as 2
```

Examples

```
awplus# configure terminal
awplus(config)# router bgp 12
awplus(config-router)#
```

```
awplus# configure terminal
awplus(config)# no router bgp 12
awplus(config)#
```

route-map (Route Map)

Use this command to configure a route map entry, and to specify whether the device will process or discard matching routes and BGP update messages.

The switch uses a name to identify the route map, and a sequence number to identify each entry in the route map.

The **route-map** command puts you into route-map configuration mode. In this mode, you can use the following:

- one or more of the **match** commands to create match clauses. These specify what routes or update messages match the entry.
- one or more of the **set** commands to create set clauses. These change the attributes of matching routes or update messages.

Use the **no** variant of this command to delete a route map or to delete an entry from a route map.

Syntax

```
route-map <mapname> {deny|permit} <seq>
no route-map <mapname>
no route-map <mapname> {deny|permit} <seq>
```

Parameter	Description
<mapname>	A name to identify the route map.
deny	The route map causes a routing process to discard matching routes or BGP update messages.
permit	The route map causes a routing process to use matching routes or BGP update messages.
<seq>	<1-65535> The sequence number of the entry. You can use this parameter to control the order of entries in this route map.

Mode Global Configuration

Usage Route maps allow you to control and modify routing information by filtering routes and setting route attributes. You can apply route maps when the device:

- processes BGP update messages that it has received from a peer
- prepares BGP update messages to send to peers
- redistributes routes from one routing protocol into another
- redistributes static routes into routing protocols
- uses BGP route flap dampening

When a routing protocol passes a route or update message through a route map, it checks the entries in order of their sequence numbers, starting with the lowest numbered entry.

If it finds a match on a route map with an action of permit, then it applies any set clauses and accepts the route. Having found a match, the route is not compared against any further entries of the route map.

If it finds a match on a route map with an action of deny, it will discard the matching route.

If it does not find a match, it discards the route or update message. This means that route maps end with an implicit deny entry. To permit all non-matching routes or update messages, end your route map with an entry that has an action of **permit** and no match clause.

Examples To enter route-map mode for entry 1 of the route map called `route1`, and then add a match and set clause to it, use the commands:

```
awplus# configure terminal
awplus(config)# route-map route1 permit 1
awplus(config-route-map)# match as-path 60
awplus(config-route-map)# set weight 70
```

To enter route-map mode for entry 2 of the route map called `route1`, and then add a match and set clause to it, use the commands:

```
awplus# configure terminal
awplus(config)# route-map route1 permit 2
awplus(config-route-map)# match interface vlan2
awplus(config-route-map)# set metric 20
```

Note how the prompt changes when you go into route map configuration mode.

To make the device process non-matching routes instead of discarding them, add a command like the following one:

```
awplus(config)# route-map route1 permit 100
```

Related Commands

- bgp dampening (BGP)**
- neighbor default-originate (BGP)**
- neighbor route-map (BGP)**
- neighbor unsuppress-map (BGP)**
- network (BGP)**
- redistribute (into BGP) (BGP)**
- show ip bgp route-map (BGP)**
- show route-map (Route Map)**

set as-path (Route Map)

Use this command to add an AS path set clause to a route map entry.

When a BGP update message matches the route map entry, the device prepends the specified Autonomous System Number (ASN) or ASNs to the update's AS path attribute.

The AS path attribute is a list of the autonomous systems through which the announcement for the prefix has passed. As prefixes pass between autonomous systems, each autonomous system adds its ASN to the beginning of the list. This means that the AS path attribute can be used to make routing decisions.

Use the **no** variant of this command to remove the set clause.

Syntax `set as-path prepend <1-65535> [<1-65535>]...`
`no set as-path prepend [<1-65535> [<1-65535>]...]`

Parameter	Description
<code>prepend</code>	Prepends the autonomous system path.
<code><1-65535></code>	The number to prepend to the AS path. If you specify multiple ASNs, separate them with spaces.

Mode Route-map mode

Usage Use the **set as-path** command to specify an autonomous system path. By specifying the length of the AS-Path, the device influences the best path selection by a neighbor. Use the `prepend` parameter with this command to prepend an AS path string to routes increasing the AS path length.

This command is valid for BGP update messages only.

Example To use entry 3 of the route map called `myroute` to prepend ASN 8 and 24 to the AS path of matching update messages, use the commands:

```
awplus# configure terminal
awplus(config)# route-map myroute permit 3
awplus(config-route-map)# set as-path prepend 8 24
```

Related Commands [match as-path \(Route Map\)](#)
[route-map \(Route Map\)](#)
[show route-map \(Route Map\)](#)

set community (Route Map)

Use this command to add a community set clause to a route map entry.

When a BGP update message matches the route map entry, the device takes one of the following actions:

- changes the update's community attribute to the specified value or values, or
- adds the specified community value or values to the update's community attribute, if you specify the **additive** parameter after specifying another parameter. or
- removes the community attribute from the update, if you specify the **none** parameter

Use the **no** variant of this command to remove the set clause.

Syntax

```
set community {[<1-65535>][AA:NN] [internet] [local-AS]
               [no-advertise] [no-export] [additive]}

no set community {[AA:NN] [internet] [local-AS] [no-advertise]
                 [no-export] [additive]}

set community none

no set community none
```

Parameter	Description
<1-65535>	The AS number of the community as an integer not in AA:NN format.
AA:NN	The Autonomous System (AS) number of the community, in AA:NN format. AS numbers are assigned to the regional registries by the IANA (www.iana.org) and can be obtained from the registry in your region. AA and NN are both integers from 1 to 65535. AA is the AS number; NN is a value chosen by the ASN administrator.
local-AS	The community of routes that must not be advertised to external BGP peers (this includes peers in other members' Autonomous Systems inside a BGP confederation).
internet	The community of routes that can be advertised to all BGP peers.
no-advertise	The community of routes that must not be advertised to other BGP peers.
no-export	The community of routes that must not be advertised outside a BGP confederation boundary (a standalone Autonomous System that is not part of a confederation should be considered a confederation itself).
none	The device removes the community attribute from matching update messages.
additive	The device adds the specified community value to the update message's community attribute, instead of replacing the existing attribute. By default this parameter is not included, so the device replaces the existing attribute.

Mode Route-map Configuration

Usage This command is valid for BGP update messages only.

Examples To use entry 3 of the route map called `rmap1` to put matching routes into the no-advertise community, use the commands:

```
awplus# configure terminal
awplus(config)# route-map rmap1 permit 3
awplus(config-route-map)# set community no-advertise
```

To use entry 3 of the route map called `rmap1` to put matching routes into several communities, use the commands:

```
awplus# configure terminal
awplus(config)# route-map rmap1 permit 3
awplus(config-route-map)# set community 10:01 23:34 12:14
no-export
```

To use entry 3 of the route map called `rmap1` to put matching routes into a single AS community numbered 16384, use the commands:

```
awplus# configure terminal
awplus(config)# route-map rmap1 permit 3
awplus(config-route-map)# set community 16384 no-export
```

Related Commands [match community \(Route Map\)](#)
[route-map \(Route Map\)](#)

show bgp ipv6 (BGP4+)

Use this command to display BGP4+ network information for a specified IPv6 address.

For information on output options, see [“Controlling “show” Command Output” on page 3.6](#).

Syntax `show bgp ipv6 <ipv6-addr>`

Parameter	Description
<code><ipv6-addr></code>	Specifies the IPv6 address, entered in hexadecimal in the format X:X::X:X.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 2001:0db8:010d::1
```

Related Commands [show bgp ipv6 longer-prefixes \(BGP4+\)](#)

show bgp ipv6 community (BGP4+)

Use this command to display routes that match specified communities within an IPv6 environment. Use the [show ip bgp community \(BGP\)](#) command within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

You may use any combination and repetition of parameters listed in the `<type>` placeholder.

Syntax `show bgp ipv6 community [<type>] [exact-match]`

Parameter	Description
<code><type></code>	{[AA:NN] [local-AS] [no-advertise] [no-export]}
<code>AA:NN</code>	Specifies the Autonomous System (AS) community number, in AA:NN format.
<code>local-AS</code>	Do not send outside local Autonomous Systems (well-known community).
<code>no-advertise</code>	Do not advertise to any peer (well-known community).
<code>no-export</code>	Do not export to next AS (well-known community).
<code>exact-match</code>	Specifies that the exact match of the communities is displayed. This optional parameter cannot be repeated.

Mode User Exec and Privileged Exec

Examples Note that the AS numbers shown are examples only.

```
awplus# show bgp ipv6 community 64497:64499 exact-match
awplus# show bgp ipv6 community 64497:64499 64500:64501
exact-match
awplus# show bgp ipv6 community 64497:64499 64500:64501
no-advertise 64510:64511
awplus# show bgp ipv6 community no-advertise no-advertise
no-advertise exact-match
awplus# show bgp ipv6 community no-export 64510:64511
no-advertise local-AS no-export
awplus# show bgp ipv6 community no-export 64510:64511
no-advertise 64497:64499 64500:64501 no-export
awplus# show bgp ipv6 community no-export 64497:64499
no-advertise local-AS no-export
```

Related Commands [show ip bgp community \(BGP\)](#)

show bgp ipv6 community-list (BGP4+)

Use this command to display routes that match the given community-list within an IPv6 environment. Use the [show ip bgp community-list \(BGP\)](#) command within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 community-list <listname> [exact-match]`

Parameter	Description
<code><listname></code>	Specifies the community list name.
<code>exact-match</code>	Displays only routes that have exactly the same specified communities.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 community-list mylist exact-match
```

Related Commands [show ip bgp community-list \(BGP\)](#)

show bgp ipv6 dampening (BGP4+)

Use this command to show dampened routes from a BGP4+ instance within an IPv6 environment. Use the [show ip bgp dampening \(BGP\)](#) command to show dampened routes from a BGP instance within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 dampening {dampened-paths|flap-statistics|parameters}`

Parameter	Description
dampened-paths	Display paths suppressed due to dampening.
flap-statistics	Display flap statistics of routes.
parameters	Display details of configured dampening parameters.

Mode User Exec and Privileged Exec

Usage Enable BGP4+ dampening to maintain dampened-path information in memory.

Examples

```
awplus# show bgp ipv6 dampening dampened-path
```

```
awplus# show bgp ipv6 dampening flap-statistics
```

```
awplus# show bgp ipv6 dampening parameter
```

Related Commands [show ip bgp dampening \(BGP\)](#)

show bgp ipv6 filter-list (BGP4+)

Use this command to display routes conforming to the filter-list within an IPv6 environment. Use the [show ip bgp filter-list \(BGP\)](#) command to display routes conforming to the filter-list within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 filter-list <listname>`

Parameter	Description
<code><listname></code>	Specifies the regular-expression access list name.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 filter-list mylist
```

Related Commands [show ip bgp filter-list \(BGP\)](#)

show bgp ipv6 inconsistent-as (BGP4+)

Use this command to display routes with inconsistent AS Paths within an IPv6 environment. Use the [show ip bgp inconsistent-as \(BGP\)](#) command to display routes with inconsistent AS paths within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 inconsistent-as`

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 inconsistent-as
```

Related Commands [show ip bgp inconsistent-as \(BGP\)](#)

show bgp ipv6 longer-prefixes (BGP4+)

Use this command to display the route of the local BGP4+ routing table for a specific prefix with a specific mask or for any prefix having a longer mask than the one specified.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 <ipv6-addr/prefix-length> longer-prefixes`

Parameter	Description
<code><ipv6-addr/prefix-length></code>	Specifies the IPv6 address with prefix length. The IPv6 address uses the format X:X::X:X/Prefix-Length. The prefix-length is usually set between 0 and 64.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 2001:0db8::/64 longer-prefixes
```

Related Commands [show bgp ipv6 \(BGP4+\)](#)

show bgp ipv6 neighbors (BGP4+)

Use this command to display detailed information on peering connections to all BGP4+ neighbors within an IPv6 environment. Use the [show ip bgp neighbors \(BGP\)](#) to display detailed information on peering connections to all BGP neighbors within an IPv4 environment. See the [Usage](#) section about what you can verify using this command.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 neighbors [<ipv6-addr> [advertised-routes|received prefix-filter|received-routes|routes]]`

Parameter	Description
<ipv6-addr>	Specifies the IPv6 address, entered in hexadecimal in the format X:X::X:X.
advertised-routes	Displays the routes advertised to a BGP4+ neighbor.
received prefix-filter	Displays received prefix-list filters.
received-routes	Displays the received routes from the neighbor. To display all the received routes from the neighbor, configure the BGP4+ soft reconfigure first.
routes	Displays all accepted routes learned from neighbors.

Mode User Exec and Privileged Exec

Usage You can verify many of the BGP4+ settings using this command:

- [show bgp ipv6 neighbors session information:](#)
- [show bgp ipv6 neighbors IPv6 unicast address family information:](#)
- [show bgp ipv6 neighbors connection information:](#)

show bgp ipv6 neighbors session information:

- Neighbor address, ASN information and if the link is external or internal
- BGP version and status
- Neighbor capabilities for the BGP session
- Number of messages transmitted and received

show bgp ipv6 neighbors IPv6 unicast address family information:

- BGP4+ table version
- IPv6 Address Family dependant capabilities
- IPv6 Communities
- IPv6 Route filters for ingress and egress updates
- Number of announced and accepted IPv6 prefixes

show bgp ipv6 neighbors connection information:

- Connection counters
- Graceful restart timer
- Hop count to the peer
- Next hop information
- Local and external port numbers

Examples [BGP4+]

```
awplus# show bgp ipv6 neighbors 2001:0db8:010d::1
advertised-routes

awplus# show bgp ipv6 neighbors 2001:0db8:010d::1 received
prefix-filter

awplus# show bgp ipv6 neighbors 2001:0db8:010d::1
received-routes

awplus# show bgp ipv6 neighbors 2001:0db8:010d::1 routes
```

Output Figure 3-5: Example output from the show bgp ipv6 neighbors 2001:db8:b::1 command

```
awplus#show bgp ipv6 neighbors 2001:db8:b::1
BGP neighbor is 2001:db8:b::1, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 2.2.2.1
  BGP state = Established, up for 01:03:26
  Last read 01:03:26, hold time is 90, keepalive interval is 30 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    4-Octet ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 157 messages, 0 notifications, 0 in queue
  Sent 228 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
  Update source is lo
For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1
  Index 2, Offset 0, Mask 0x4
  Community attribute sent to this neighbor (both)
  0 accepted prefixes
  0 announced prefixes

For address family: IPv6 Unicast
  BGP table version 66, neighbor version 66
  Index 2, Offset 0, Mask 0x4
  AF-dependant capabilities:
    Graceful restart: advertised, received

  Community attribute sent to this neighbor (both)
  Default information originate, default sent
  Inbound path policy configured
  Incoming update prefix filter list is *BGP_FILTER_LIST
  Route map for incoming advertisements is *BGP_LOCAL_PREF_MAP
  8 accepted prefixes
  8 announced prefixes

Connections established 1; dropped 0
Graceful-restart Status:
  Remote restart-time is 90 sec

  External BGP neighbor may be up to 2 hops away.
  Local host: 2001:db8:a::1, Local port: 179
  Foreign host: 2001:db8:b::1, Foreign port: 50672
  Nexthop: 1.1.1.1
  Nexthop global: 2001:db8:a::1
  Nexthop local: ::
  BGP connection: non shared network
```

Related Commands [show ip bgp neighbors \(BGP\)](#)

show bgp ipv6 paths (BGP4+)

Use this command to display BGP4+ path information within an IPv6 environment. Use the [show ip bgp paths \(BGP\)](#) command to display BGP path information within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 paths`

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 paths
```

Related Commands [show ip bgp paths \(BGP\)](#)

show bgp ipv6 prefix-list (BGP4+)

Use this command to display routes matching the prefix-list within an IPv6 environment. Use the [show ip bgp prefix-list \(BGP\)](#) command to display routes matching the prefix-list within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 prefix-list <list>`

Parameter	Description
<code><list></code>	Specifies the name of the IPv6 prefix list.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 prefix-list mylist
```

Related Commands [show ip bgp prefix-list \(BGP\)](#)

show bgp ipv6 quote-regexp (BGP4+)

Use this command to display routes matching the AS path regular expression within an IPv6 environment. Use the [show ip bgp quote-regexp \(BGP\)](#) command to display routes matching the AS path regular expression within an IPv4 environment.

Note that you must use quotes to enclose the regular expression with this command. Use the regular expressions listed below with the *<expression>* parameter:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 quote-regexp <expression>`

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 quote-regexp myexpression
```

Related Commands [show ip bgp quote-regexp \(BGP\)](#)

show bgp ipv6 regexp (BGP4+)

Use this command to display routes matching the AS path regular expression within an IPv6 environment. Use the [show ip bgp regexp \(BGP\)](#) command to display routes matching the AS path regular expression within an IPv4 environment.

Use the regular expressions listed below with the *<expression>* parameter:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 regexp <expression>`

Parameter	Description
<i><expression></i>	Specifies a regular-expression to match the BGP4+ AS paths.

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 regexp myexpression
```

Related Commands [show ip bgp regexp \(BGP\)](#)

show bgp ipv6 route-map (BGP4+)

Use this command to display BGP4+ routes that match the specified route-map within an IPv6 environment. Use the [show ip bgp route-map \(BGP\)](#) command to display BGP routes that match the specified route-map within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 route-map <route-map>`

Parameter	Description
<code><route-map></code>	Specifies a route-map that is matched.

Mode User Exec and Privileged Exec

Example To show routes that match the route-map `myRouteMap`, use the command:

```
awplus# show bgp ipv6 route-map myRouteMap
```

Related Commands [show ip bgp route-map \(BGP\)](#)

show bgp ipv6 summary (BGP4+)

Use this command to display a summary of a BGP4+ neighbor status within an IPv6 environment. Use the [show ip bgp summary \(BGP\)](#) command to display a summary of a BGP neighbor status within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show bgp ipv6 summary`

Mode User Exec and Privileged Exec

Example

```
awplus# show bgp ipv6 summary
```

Related Commands [show ip bgp summary \(BGP\)](#)

show bgp nexthop-tracking (BGP)

Use this command to display BGP nexthop-tracking status.

Syntax `show bgp nexthop-tracking`

Mode User Exec and Privileged Exec

Example To display BGP nexthop-tracking status, use the command:

```
awplus# show bgp nexthop-tracking
```

Related Commands [bgp nexthop-trigger-count \(BGP and BGP4+\)](#)
[show bgp nexthop-tree-details \(BGP\)](#)

show bgp nexthop-tree-details (BGP)

Use this command to display BGP nexthop-tree-details.

Syntax `show bgp nexthop-tree-details`

Mode User Exec and Privileged Exec

Example To display BGP nexthop-tree-details, use the command:

```
awplus# show bgp nexthop-tree-details
```

Related Commands [show bgp nexthop-tracking \(BGP\)](#)

show bgp memory maxallocation (BGP)

This command displays the maximum percentage of total memory that is allocated to BGP processes.

For information on output options, see [“Controlling “show” Command Output” on page 3.6](#).

Syntax `show bgp memory maxallocation`

Mode User Exec and Privileged Exec

Example To display the maximum amount of memory allocated for BGP processes, use the command:

```
awplus# show bgp memory maxallocation
```

Output **Figure 3-6: Example output from the show bgp memory maxallocation command**

```
BGP maximum RAM allocation is 100%
```

show debugging bgp (BGP)

Use this command to display the BGP debugging option set.

For information on output options, see [“Controlling “show” Command Output” on page 3.6](#).

Syntax show debugging bgp

Mode User Exec and Privileged Exec

Example

```
awplus# show debugging bgp
```

Output **Figure 3-7: Example output from the show debugging bgp command**

```
BGP debugging status:
  BGP debugging is on
  BGP events debugging is on
  BGP updates debugging is on
  BGP fsm debugging is on
```

Related Commands [debug bgp \(BGP\)](#)

show ip bgp (BGP)

Use this command to display BGP network information.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp [*<ip-addr>* | *<ip-addr/m>*]

Parameter	Description
<i><ip-addr></i>	Specifies the IPv4 address and the optional prefix mask length.
<i><ip-addr/m></i>	

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp 10.10.1.34/24
```

Output **Figure 3-8: Example output from the show ip bgp command**

```
BGP table version is 7, local router ID is 80.80.80.80
Status codes: s suppressed, d damped, h history, * valid, >
best, i - internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight
Path
S>i10.70.0.0/24     192.10.23.67      0      100      0 ?
S>i30.30.30.30/32   192.10.23.67      0      100      0 ?
S>i63.63.63.1/32    192.10.23.67      0      100      0 ?
S>i67.67.67.67/32   192.10.23.67      0      100      0 ?
S>i172.22.10.0/24   192.10.23.67      0      100      0 ?
S>i192.10.21.0      192.10.23.67      0      100      0 ?
S>i192.10.23.0      192.10.23.67      0      100      0 ?

Total number of prefixes 7
```

Related Commands [neighbor remove-private-AS \(BGP\)](#)

show ip bgp attribute-info (BGP)

Use this command to show internal attribute hash information.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp attribute-info

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp attribute-info
```

Output **Figure 3-9: Example output from the show ip bgp attribute-info command**

```
attr[1] nexthop 0.0.0.0
attr[1] nexthop 10.10.10.10
attr[1] nexthop 10.10.10.50
```

show ip bgp cidr-only (BGP)

Use this command to display routes with non-natural network masks.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp cidr-only

Syntax [VRF] show ip bgp [global|vrf <vrf-name>] cidr-only

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp cidr-only
awplus# show ip bgp vrf red cidr-only
```

Output **Figure 3-10: Example output from the show ip bgp cidr-only command**

```
BGP table version is 0, local router ID is 10.10.10.50
Status codes: s suppressed, d damped, h history, p stale, *
valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network             Next Hop             Metric LocPrf Weight
Path
*> 3.3.3.0/24          10.10.10.10                    0 11
i
*> 6.6.6.0/24          0.0.0.0                      32768 i
Total number of prefixes 2
```

show ip bgp community (BGP)

Use this command to display routes that match specified communities from a BGP instance within an IPv4 environment. Use the [show bgp ipv6 community \(BGP4+\)](#) command within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

You may use any combination and repetition of parameters listed in the `<type>` placeholder.

Syntax `show ip bgp community [<type>] [exact-match]`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] community [<type>] [exact-match]`

Parameter	Description
<code>global</code>	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
<code>vrf</code>	Apply the command to the specified VRF instance.
<code><vrf-name></code>	The name of the VRF instance.
<code><type></code>	{[AA:NN] [local-AS] [no-advertise] [no-export]}
<code>AA:NN</code>	Specifies the Autonomous System (AS) community number, in AA:NN format.
<code>local-AS</code>	Do not send outside local Autonomous Systems (well-known community).
<code>no-advertise</code>	Do not advertise to any peer (well-known community).
<code>no-export</code>	Do not export to next AS (well-known community).
<code>exact-match</code>	Specifies that the exact match of the communities is displayed. This optional parameter cannot be repeated.

Mode User Exec and Privileged Exec

Examples Note that the AS numbers shown are examples only.

```
awplus# show ip bgp community 64497:64499 exact-match
awplus# show ip bgp community 64497:64499 64500:64501
exact-match
awplus# show ip bgp community 64497:64499 64500:64501
no-advertise 64510:64511
awplus# show ip bgp community no-advertise no-advertise
no-advertise exact-match
awplus# show ip bgp community no-export 64510:64511
no-advertise local-AS no-export
```

```
awplus# show ip bgp community no-export 64510:64511
       no-advertise 64497:64499 64500:64501 no-export

awplus# show ip bgp community no-export 64497:64499
       no-advertise local-AS no-export

awplus# show ip bgp vrf red no-export

awplus# show ip bgp global 65500:2 65500:3 exact-match
```

Related Commands [set community \(Route Map\)](#)
[show bgp ipv6 community \(BGP4+\)](#)

show ip bgp community-info (BGP)

Use this command to list all BGP community information.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp community-info

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp community-info
```

show ip bgp community-list (BGP)

Use this command to display routes that match the given community-list from a BGP instance within an IPv4 environment. Use the [show bgp ipv6 community-list \(BGP4+\)](#) command within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp community-list <listname> [exact-match]`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] community-list <listname> [exact-match]`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<listname>	Specifies the community list name.
exact-match	Displays only routes that have exactly the same specified communities.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp community-list mylist exact-match
awplus# show ip bgp vrf red community-list myCommunity
awplus# show ip bgp global community-list myExactCommunity
exact-match
```

Related Commands [show bgp ipv6 community-list \(BGP4+\)](#)

show ip bgp dampening (BGP)

Use this command to show dampened routes from a BGP instance within an IPv4 environment. Use the [show bgp ipv6 dampening \(BGP4+\)](#) command within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp dampening {dampened-paths|flap-statistics|parameters}`

Syntax [VRF] `show ip bgp [unicast] dampening {dampened-paths|flap-statistics|parameters}`

`show ip bgp [global|vrf <vrf-name>] dampening {dampened-paths|flap-statistics|parameters}`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
dampened-paths	Display paths suppressed due to dampening.
flap-statistics	Display flap statistics of routes.
parameters	Display details of configured dampening parameters.

Mode User Exec and Privileged Exec

Usage Enable BGP dampening to maintain dampened-path information in memory.

Examples

```
awplus# show ip bgp dampening dampened-paths
awplus# show ip bgp vrf red dampening dampened-paths
awplus# show ip bgp global dampening flap-statistics
```

Output **Figure 3-11: Example output from the show ip bgp dampening command**

```
dampening 15 750 2000 60 15
Reachability Half-Life time      : 15 min
Reuse penalty                    : 750
Suppress penalty                 : 2000
Max suppress time                : 60 min
Un-reachability Half-Life time   : 15 min
Max penalty (ceil)              : 11999
Min penalty (floor)             : 375
```

The following example output shows that the internal route (i), has flapped 3 times and is now categorized as history (h).

Figure 3-12: Example output from the show ip bgp dampening flap-statistics command

```
awplus# show ip bgp dampening flap-statistics
BGP table version is 1, local router ID is 30.30.30.77
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S
Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          From                Flaps  Duration  Reuse    Path
  hi1.1.1.0/24     10.100.0.62         3     00:01:20      i
```

The following example output shows a dampened route in the 1.1.1.0/24 network.

Figure 3-13: Example output from the show ip bgp dampening dampened-path command

```
awplus# show ip bgp dampening dampened-paths
BGP table version is 1, local router ID is 30.30.30.77
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S
Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          From                Reuse    Path
  di 1.1.1.0/24    10.100.0.62         00:35:10  i
Total number of prefixes 1
```

Related Commands [show bgp ipv6 dampening \(BGP4+\)](#)

show ip bgp filter-list (BGP)

Use this command to display routes conforming to the filter-list within an IPv4 environment. Use the [show bgp ipv6 filter-list \(BGP4+\)](#) command to display routes conforming to the filter-list within an IPv6 environment

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp filter-list <listname>

Syntax [VRF] show ip bgp [global|vrf <vrf-name>] filter-list <listname>

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<listname>	Specifies the regular-expression access list name.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp filter-list mylist
awplus# show ip bgp vrf red filter-list mylist
```

Related Commands [show bgp ipv6 filter-list \(BGP4+\)](#)

show ip bgp inconsistent-as (BGP)

Use this command to display routes with inconsistent AS Paths within an IPv4 environment. Use the [show bgp ipv6 inconsistent-as \(BGP4+\)](#) command to display routes with inconsistent AS paths within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp inconsistent-as`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] inconsistent-as`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp inconsistent-as
```

```
awplus# show ip bgp global inconsistent-as
```

Related Commands [show bgp ipv6 inconsistent-as \(BGP4+\)](#)

show ip bgp longer-prefixes (BGP)

Use this command to display the route of the local BGP routing table for a specific prefix with a specific mask, or for any prefix having a longer mask than the one specified.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp <ip-address/m> longer-prefixes`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] <ip-address/m> longer-prefixes`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<ip-address/m>	Neighbor's IP address and subnet mask, entered in the form A.B.C.D/M. Where M is the subnet mask length.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp 10.10.0.10/24 longer-prefixes
awplus# show ip bgp vrf red 172.16.4.0/24
awplus# show ip bgp global 172.16.0.0/16 longer-prefixes
```

show ip bgp neighbors (BGP)

Use this command to display detailed information on peering connections to all BGP neighbors within an IPv4 environment. Use the [show bgp ipv6 neighbors \(BGP4+\)](#) to display detailed information on peering connections to all BGP4+ neighbors within an IPv6 environment. See the [Usage](#) section about what you can verify using this command.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax [BGP] `show ip bgp neighbors [<ipv4-addr> [advertised-routes|received prefix-filter|received-routes|routes]]`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] neighbors [<ipv4-addr> routes]`

Parameter	Description
<ipv4-addr>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.
advertised-routes	Displays the routes advertised to a BGP neighbor.
received prefix-filter	Displays the received prefix-list filters.
received-routes	Displays the received routes from the neighbor. To display all the received routes from the neighbor, configure the BGP soft reconfigure first.
routes	Displays all accepted routes learned from neighbors.
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode [BGP] User Exec and Privileged Exec

Usage [BGP] You can verify many of the BGP settings using this command:

- **show bgp ip neighbors session information:**
- **show bgp ip neighbors IPv4 unicast address family Information:**
- **show bgp ip neighbors connection information:**

show bgp ip neighbors session information:

- Neighbor address, ASN information and if the link is external or internal
- BGP version and status
- Neighbor capabilities for the BGP session
- Number of messages transmitted and received

show bgp ip neighbors IPv4 unicast address family Information:

- BGP table version
- IPv4 Address Family dependant capabilities
- IPv4 Communities
- IPv4 Route filters for ingress and egress updates
- Number of announced and accepted IPv4 prefixes

show bgp ip neighbors connection information:

- Connection counters
- Graceful restart timer
- Hop count to the peer
- Next hop information
- Local and external port numbers

Examples [BGP]

```
awplus# show ip bgp neighbors 10.10.10.72 advertised-routes
awplus# show ip bgp neighbors 10.10.10.72 received
prefix-filter
awplus# show ip bgp neighbors 10.10.10.72 received-routes
awplus# show ip bgp neighbors 10.10.10.72 routes
```

Output **Figure 3-14: Example output from the show ip bgp neighbors 10.10.10.72 command**

```
awplus#show ip bgp neighbors 10.10.10.72
BGP neighbor is 10.10.10.72, remote AS 100, local AS 100, internal
link
Member of peer-group group1 for session parameters
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active
  Last read      , hold time is 90, keepalive interval is 30 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 5 seconds
For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 1, Offset 0, Mask 0x2
  group1 peer-group member
  NEXT_HOP is always this router
  0 accepted prefixes
  0 announced prefixes

Connections established 0; dropped 0
Next connect timer due in 33 seconds
```

Related Commands **show bgp ipv6 neighbors (BGP4+)**

show ip bgp neighbors connection-retrytime (BGP)

Use this command to display the configured connection-retrytime value of the peer at the session establishment time with the neighbor.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr>
connection-retrytime`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 connection-retrytime
```

show ip bgp neighbors hold-time (BGP)

Use this command to display the configured holdtime value of the peer at the session establishment time with the neighbor.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> hold-time`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Default The holdtime timer default is 90 seconds as per RFC 4271. Holdtime is keepalive * 3.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp neighbors 10.11.4.26 hold-time
```

Related Commands [neighbor timers \(BGP and BGP4+\)](#)
[show ip bgp neighbors keepalive-interval \(BGP\)](#)
[timers \(BGP and BGP4+\)](#)

show ip bgp neighbors keepalive (BGP)

Use this command to display the number of keepalive messages sent to the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> keepalive`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp neighbors 10.11.4.26 keepalive
```

Related Commands [show ip bgp neighbors keepalive-interval \(BGP\)](#)

show ip bgp neighbors keepalive-interval (BGP)

Use this command to display the configured keepalive-interval value of the peer at the session establishment time with the neighbor.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors {<ipv4-addr> | <ipv6-addr>} keepalive-interval`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Default The keepalive timer default is 60 seconds as per RFC 4271. Keepalive is holdtime / 3.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp neighbors 10.11.4.26 keepalive-interval
```

Related Commands [neighbor timers \(BGP and BGP4+\)](#)
[show ip bgp neighbors hold-time \(BGP\)](#)
[timers \(BGP and BGP4+\)](#)

show ip bgp neighbors notification (BGP)

Use this command to display the number of notification messages sent to the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors {<ipv4-addr> | <ipv6-addr>} notification`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 notification
```

show ip bgp neighbors open (BGP)

Use this command to display the number of open messages sent to the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> open`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 open
```

show ip bgp neighbors rcvd-msgs (BGP)

Use this command to display the number of messages received by the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> rcvd-msgs`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 rcvd-msgs
```

show ip bgp neighbors sent-msgs (BGP)

Use this command to display the number of messages sent to the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> sent-msgs`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 sent-msgs
```

show ip bgp neighbors update (BGP)

Use this command to display the number of update messages sent to the neighbor from the peer throughout the session.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp neighbors <ipv4-addr> update`

Parameter	Description
<code><ipv4-addr></code>	The IPv4 address of an IPv4 BGP neighbor, in dotted decimal notation A.B.C.D.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp neighbors 10.11.4.26 update
```

show ip bgp paths (BGP)

Use this command to display BGP4 path information within an IPv4 environment. Use the [show bgp ipv6 paths \(BGP4+\)](#) command to display BGP4+ path information within an IPv4 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp paths`

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp paths
```

Related Commands [show bgp ipv6 paths \(BGP4+\)](#)

show ip bgp prefix-list (BGP)

Use this command to display routes matching the prefix-list within an IPv4 environment. Use the [show bgp ipv6 prefix-list \(BGP4+\)](#) command to display routes matching the prefix-list within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp prefix-list <list>`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] prefix-list <list>`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<list>	Specifies the name of the IP prefix list.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp prefix-list mylist
```

```
awplus# show ip bgp vrf red prefix-list myPrefixes
```

Related Commands [show bgp ipv6 prefix-list \(BGP4+\)](#)

show ip bgp quote-regexp (BGP)

Use this command to display routes matching the AS path regular expression within an IPv4 environment. Use the [show bgp ipv6 quote-regexp \(BGP4+\)](#) command to display routes matching the AS path regular expression within an IPv6 environment.

Note that you must use quotes to enclose the regular expression with this command. Use the regular expressions listed below with the *<expression>* parameter:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp quote-regexp <expression>`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] quote-regexp <expression>`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<expression>	Specifies a regular-expression to match the BGP AS paths.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp quote-regexp myexpression
```

```
awplus# show ip bgp global quote-regexp 65550 65555
```

Related Commands [show bgp ipv6 quote-regexp \(BGP4+\)](#)

show ip bgp regexp (BGP)

Use this command to display routes matching the AS path regular expression within an IPv4 environment. Use the [show bgp ipv6 regexp \(BGP4+\)](#) command to display routes matching the AS path regular expression within an IPv6 environment.

Use the regular expressions listed below with the *<expression>* parameter

Use the regular expressions listed below with the *<expression>* parameter:

Symbol	Character	Meaning
^	Caret	Used to match the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
\$	Dollar sign	Used to match the end of the input string.
.	Period	Used to match a single character (white spaces included).
*	Asterisk	Used to match none or more sequences of a pattern.
+	Plus sign	Used to match one or more sequences of a pattern.
?	Question mark	Used to match none or one occurrence of a pattern.
_	Underscore	Used to match spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	Specifies a range of single-characters.
-	Hyphen	Separates the end points of a range.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp regexp <expression>`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] regexp <expression>`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<expression>	Specifies a regular-expression to match the BGP AS paths.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp regexp myexpression
```

```
awplus# show ip bgp vrf red regexp 65550 65555
```

Related Commands [show bgp ipv6 regexp \(BGP4+\)](#)

show ip bgp route-map (BGP)

Use this command to display BGP routes that match the specified route-map within an IPv4 environment. Use the [show bgp ipv6 route-map \(BGP4+\)](#) command to display BGP4+ routes that match the specified route-map within an IPv6 environment.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp route-map <route-map>`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] route-map <route-map>`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.
<route-map>	Specifies a route-map that is matched.

Mode User Exec and Privileged Exec

Examples To show routes that match the route-map `myRouteMap` for the global routing instance, use the command:

```
awplus# show ip bgp global route-map myRouteMap
```

To show routes that match the route-map `myRouteMap`, use the command:

```
awplus# show ip bgp route-map myRouteMap
```

Related Commands [show bgp ipv6 route-map \(BGP4+\)](#)

show ip bgp scan (BGP)

Use this command to display BGP scan status.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip bgp scan

Mode User Exec and Privileged Exec

Example

```
awplus# show ip bgp scan
```

Output **Figure 3-15: Example output from the show ip bgp scan command**

```
BGP scan is running
BGP scan interval is 60
BGP instance : AS is 11,DEFAULT
Current BGP nexthop cache:
BGP connected route:
 10.10.10.0/24
 10.10.11.0/24
```

show ip bgp summary (BGP)

Use this command to display a summary of a BGP neighbor status within an IPv4 environment. Use the [show bgp ipv6 summary \(BGP4+\)](#) command to display a summary of BGP4+ neighbors.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip bgp summary`

Syntax [VRF] `show ip bgp [global|vrf <vrf-name>] summary`

Parameter	Description
global	When VRF-Lite is configured, apply the command to the global routing and forwarding table.
vrf	Apply the command to the specified VRF instance.
<vrf-name>	The name of the VRF instance.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip bgp summary
```

```
awplus# show ip bgp vrf red summary
```

Output **Figure 3-16: Example output from the show ip bgp summary command**

```
awplus>show ip bgp summary
BGP router identifier 0.0.0.0, local AS number 100
BGP table version is 10
BGP AS-PATH entries 0
BGP community entries
Neighbor          V      AS   MsgRc  MsgSnt  TblVer  InOutQ  Up/Down  State/PfxRcd
10.10.10.72       4      100    0      0       0 0/0    never     Active
2001:0db8:010d::1 4        1     0      0       0 0/0    never     Active
Number of neighbors 2
```

Related Commands [show bgp ipv6 summary \(BGP4+\)](#)

show ip community-list (BGP and BGP4+)

Use this command to display routes that match a specified community-list name or number.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip community-list [<listnumber>|<listname>]`

Parameter	Description
<code><listnumber></code>	Specifies the community list number in the range <1-199> as specified by a previously issued ip community-list command.
<code><listname></code>	Specifies the community list name as specified by a previously issued ip community-list (BGP and BGP4+) command.

Mode User Exec and Privileged Exec

Examples

```
awplus# show ip community-list mylist
awplus# show ip community-list 99
```

Related Commands [ip community-list \(BGP and BGP4+\)](#)
[ip community-list expanded \(BGP and BGP4+\)](#)
[ip community-list standard \(BGP and BGP4+\)](#)

show ip extcommunity-list (BGP and BGP4+)

Use this command to display a configured extcommunity-list.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax `show ip extcommunity-list [<1-199>|<extcommunity-listname>]`

Parameter	Description
<code><1-199></code>	Extcommunity-list number
<code><extcommunity-listname></code>	Extcommunity-list name

Mode User Exec and Privileged Exec

Example

```
awplus# show ip extcommunity-list 33
```

Related Commands [ip extcommunity-list expanded \(BGP and BGP4+\)](#)
[ip extcommunity-list standard \(BGP and BGP4+\)](#)

show ip prefix-list (IPv4 Prefix List)

Use this command to display the IPv4 prefix-list entries. Note that this command is valid for RIP and BGP routing protocols only.

Syntax `show ip prefix-list [<name>|detail|summary]`

Parameter	Description
<name>	Specify the name of a prefix list in this placeholder.
detail	Specify this parameter to show detailed output for all IPv4 prefix lists.
summary	Specify this parameter to show summary output for all IPv4 prefix lists.

Mode User Exec and Privileged Exec

Example

```
awplus# show ip prefix-list
awplus# show ip prefix-list 10.10.0.98/8
awplus# show ip prefix-list detail
```

Related Commands [ip prefix-list \(IPv4 Prefix List\)](#)

show ipv6 prefix-list (IPv6 Prefix List)

Use this command to display the prefix-list entries. Note that this command is valid for RIPng and BGP4+ routing protocols only.

Syntax `show ipv6 prefix-list [<name>|detail|summary]`

Parameter	Description
<name>	Specify the name of an individual IPv6 prefix list.
detail	Specify this parameter to show detailed output for all IPv6 prefix lists.
summary	Specify this parameter to show summary output for all IPv6 prefix lists.

Mode User Exec and Privileged Exec

Example

```
awplus# show ipv6 prefix-list
awplus# show ipv6 prefix-list 10.10.0.98/8
awplus# show ipv6 prefix-list detail
```

Related Commands [ipv6 prefix-list \(IPv6 Prefix List\)](#)

show ip protocols bgp (BGP)

Use this command to display BGP process parameters and statistics.

For information on output options, see [“Controlling “show” Command Output” on page 3.6.](#)

Syntax show ip protocols bgp

Mode User Exec and Privileged Exec

Example To display BGP process parameters and statistics, use the command:

```
awplus# show ip protocols bgp
```

Output

```
Routing Protocol is "bgp 100"
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Default local-preference applied to incoming route is 100
  Redistributing:
  Neighbor(s):
  Address AddressFamily FiltIn FiltOut DistIn DistOut RouteMapIn RouteMapOut
  Weight
  10.10.10.1                unicast
```

show route-map (Route Map)

Use this command to display information about one or all route maps.

Syntax `show route-map <map-name>`

Parameter	Description
<code><map-name></code>	A name to identify the route map.

Mode User Exec and Privileged Exec

Example To display information about the route-map named `example-map`, use the command:

```
awplus# show route-map example-map
```

Output **Figure 3-17: Example output from the show route-map command**

```
route-map example-map, permit, sequence 1
  Match clauses:
    ip address prefix-list example-pref
  Set clauses:
    metric 100
route-map example-map, permit, sequence 200
  Match clauses:
  Set clauses:
```

Related Commands [route-map \(Route Map\)](#)

synchronization (BGP and BGP4+)

Use this command in Router Configuration mode or in Address Family Configuration mode to ensure BGP does not advertise routes learned from iBGP peers until they are learned locally, or are propagated throughout the AS via an IGP.

Use the **no** variant of this command to disable this function.

Syntax `synchronization`
`no synchronization`

Default Disabled.

Mode Router Configuration and Address Family Configuration mode

Usage Synchronization is used when a BGP router should not advertise routes learned from iBGP neighbors, unless those routes are also present in an IGP (for example, OSPF). These routes must be in the RIB (Routing Information Base) learned locally or via an IGP.

Synchronization may be enabled when all the routers in an autonomous system do not speak BGP, and the autonomous system is a transit for other autonomous systems.

Use the **no synchronization** command when BGP router can advertise routes learned from iBGP neighbors, without waiting for IGP reachability, when routes are in the RIB.

Example The following example enables IGP synchronization of iBGP routes in Router Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config-router)# synchronization
```

The following example enables IGP synchronization of iBGP routes in IPv4 unicast Address Family Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config)# address-family ipv4 unicast
awplus(config-af)# synchronization
```

The following example enables IGP synchronization of iBGP routes in the IPv6 unicast Address Family Configuration mode:

```
awplus# configure terminal
awplus(config)# router bgp 11
awplus(config)# address-family ipv6 unicast
awplus(config-af)# synchronization
```

timers (BGP and BGP4+)

Use this command sets the BGP keepalive timer and holdtime timer values.

Use the **no** variant of this command to reset timers to the default.

Syntax `timers bgp <keepalive> <holdtime>`
`no timers bgp [<keepalive> <holdtime>]`

Parameter	Description
<code><keepalive></code>	<p><code><0-65535></code> The frequency with which the keepalive messages are sent to the neighbors. The default is 30 seconds as per RFC 4271.</p> <p>Cisco IOS uses a 60 second keepalive timer default value. Adjust keepalive timers for interoperability as required. Maintain the keepalive value at the holdtime value / 3.</p>
<code><holdtime></code>	<p><code><0-65535></code> The interval after which the neighbor is considered dead if keepalive messages are not received. The default holdtime value is 90 seconds as per RFC 4271.</p> <p>Cisco IOS uses a 180 second holdtime timer default value. Adjust holdtime timers for interoperability as required. Maintain the holdtime value at the keepalive value * 3.</p>

Default The keepalive timer default is 60 seconds, the holdtime timer default is 90 seconds, and the connect timer default is 120 seconds as per RFC 4271. Holdtime is keepalive * 3.

Mode Router Configuration

Usage This command is used globally to set or unset the keepalive and holdtime values for all the neighbors.

Examples

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# timers bgp 40 120
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no timers bgp 30 90
```

```
awplus# configure terminal
awplus(config)# router bgp 10
awplus(config-router)# no timers bgp
```

Related Commands [neighbor timers \(BGP and BGP4+\)](#)
[show ip bgp neighbors hold-time \(BGP\)](#)
[show ip bgp neighbors keepalive-interval \(BGP\)](#)

undebg bgp (BGP)

Use this command to disable BGP debugging functions.

Syntax `undebg bgp [all|dampening|events|filters|fsm|keepalives|nht|nsm|updates]`

`undebg all bgp`

Parameter	Description
all	Disable all debugging for BGP.
dampening	Disable debugging for BGP dampening.
events	Disable debugging for BGP events.
filters	Disable debugging for BGP filters.
fsm	Disable debugging for BGP Finite State Machine (FSM).
keepalives	Disable debugging for BGP keepalives.
nht	Disable debugging for BGP NHT (Next Hop Tracking) messages.
nsm	Disable debugging for NSM messages.
updates	Disable debugging for BGP updates.

Mode Privileged Exec and Global Configuration

Example

```
awplus# undebg bgp events
awplus# undebg bgp nht
awplus# undebg bgp updates
```

Related Commands [debug bgp \(BGP\)](#)

