Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP) exchanges routing information among neighboring routers in different Autonomous Systems (AS). Arista switches use BGP version 4+, incorporating the multiprotocol extensions defined by RFC 4760 so that BGP can carry both IPv4 and IPv6 routes simultaneously over a single BGP peering.

This chapter contains the following sections.

- Section 33.1: BGP Conceptual Overview
- Section 33.2: Configuring BGP
- Section 33.3: BGP Examples
- Section 33.4: BGP Commands

Arista switches support these BGP functions:

- A single BGP instance
- Simultaneous internal (iBGP) and external (eBGP) peering
- Multiprotocol BGP
- BGP confederations
33.1 BGP Conceptual Overview

BGP is a protocol that exchanges routing information among neighboring routers in different autonomous systems through TCP sessions.

BGP neighbors (peers) communicate through a TCP session on port 179. They are established by manual configuration commands (static peers) or by creating a peer group listen range and accepting incoming peering requests in that range (dynamic peers). Internal BGP (iBGP) peers operate within a single autonomous system (AS). External BGP (eBGP) peers operate between autonomous systems. Border routers are on AS boundaries and exchange information with other autonomous systems; the primary function of border routers is distributing routes. Internal routers do not distribute route updates that they receive.

BGP defines a state machine for establishing connections. BGP routers maintain a state variable for each peer-to-peer session to track connection status. The state machine consists of these states:

- **Idle**: The router initializes BGP resources, refuses inbound BGP connection attempts, initiates a TCP connection to the peer, then transitions to the **Connect** state.

- **Connect**: The router waits for the TCP connection to complete, then sends an OPEN message to the peer and transitions to the **OpenSent** state if successful. If unsuccessful, it sets the **ConnectRetry** timer and transitions to the **Active** state upon expiry.

- **Active**: The router sets the **ConnectRetry** timer to zero and returns to the **Connect** state.

- **OpenSent**: The router waits for an OPEN message from the peer. After receiving a valid message, it transitions to the **OpenConfirm** state.

- **OpenConfirm**: The router waits for a keepalive message from its peer. If the message is received prior to a timeout expiry, the router transitions to the **Established** state. If the timeout expires or an error condition exists, the router transitions to the **Idle** state.

- **Established**: Peers exchange UPDATE messages about routes they advertise. If an UPDATE message contains an error, the router sends a NOTIFICATION message and transitions to the **Idle** state.

During established BGP sessions, routers exchange UPDATE messages about the destinations to which they offer connectivity. The route description includes the destination prefix, prefix length, autonomous systems in the path, the next hop, and information that affects the acceptance policy of the receiving router. UPDATE messages also list destinations to which the router no longer offers connectivity.

BGP detects and eliminates routing loops while making routing policy decisions by using the network topology as defined by AS paths and path attributes.

33.1.1 Multiprotocol BGP

Multiprotocol BGP facilitates the advertisement of network routes and switch capabilities to neighbors from multiple address families over a single BGP peering. The switch supports IPv4 unicast and IPv6 unicast address families.

Neighbors negotiate to select an address family when establishing a connection. The peer session is based on this address family, which identifies the following:

- The set of network layer protocols to which the address carried in the Next Hop field must belong.
- The encoding format of the next hop address.
- The semantics of Network Layer Reachability Information (NLRI).
33.1.2 BGP Confederations

BGP confederations divide an autonomous system (AS) into subsystems (sub-ASs), each identified by a unique sub-AS number, while still appearing externally as a single AS.

33.1.3 QoS Control of Neighbor Discovery and ARP Packets

To help prevent BGP sessions from being affected by dropped neighbor discovery and ARP packets, some Arista switches assign those packets to a higher priority output queue when they are being software forwarded. This helps minimize hardware drops from competition with data plane packets traffic congestion.

The switch platforms which use this feature are:
- DCS-7500E
- DCS-7250X
- DCS-7300X
- DCS-7010X
- DCS-7050X

33.1.4 Best-Path Selection

Routing information received via the BGP protocol often contains more than one route to the same destination: the BGP best-path selection algorithm determines which of these routes will be installed in the routing table. The following criteria are evaluated in order; at each step, if there is a tie for best path the next criterion is applied. If there is still a tie at the end of the process, BGP installs the route received from the peer with the lowest address. When equal cost multi path (ECMP) routing is enabled, multiple paths to a single destination may be installed in the IP routing table.

Route preferences can be shaped through configuration choices as described in Configuring Best-path Selection. The steps in Arista’s best-path selection process are described below.

Step 1 Select the route that has the highest weight value. The weight of paths learned from a specific neighbor can be adjusted using the neighbor weight or neighbor route-map (BGP) command. Weight is only significant locally, and is not communicated in update messages.

Step 2 Select the route that has the highest LOCAL_PREF value.

Step 3 If bgp bestpath as-path ignore is disabled (the default), select the route that has the shortest AS path, excluding confederation segments; otherwise skip ahead to the next step.

Step 4 Select the route that has the lower ORIGIN value. The lowest ORIGIN value is IGP (Interior Gateway Protocol), which is better than EGP (Exterior Gateway Protocol), which is better than INCOMPLETE.

Note Exterior Gateway Protocol (EGP), as described in RFC 904, is the predecessor of BGP and is included in the selection algorithm for legacy reasons. It is rarely if ever used, and should not be confused with eBGP (see step 6 below).

Step 5 Select the route that has the lower MED value. The MED values are comparable if bgp always-compare-med is enabled or if the routes have the same neighbor AS. The neighbor AS of a path is determined as follows:

a. If the router is in a confederation: if bgp bestpath med confed is configured, the confederation ID is used for the neighbor AS. Otherwise, the MED values of the routes are not comparable.
If the router is not in a confederation: the first AS number in the first AS_SEQUENCE segment is used as the neighbor AS. If such a segment does not exist, the local router AS number is used.

Step 6  Prefer routes received from external peers (eBGP routes) over those received from internal peers (iBGP routes).

Step 7  Select the route that has the lowest IGP cost to the BGP NEXT_HOP. The IGP cost is determined using the metric of the resolving route for the BGP next-hop.

Step 8  If `bgp bestpath as-path multipath-relax` is enabled (default), then skip ahead to the next step; otherwise the AS_PATH path lengths are compared. The path length is determined by adding the total number of AS and segment types in the AS_PATH field. Prefer the route with the shorter path length. If path lengths are equal, then prefer the route with the first smaller non-matching AS, and then prefer the route with the first smaller non-matching segment type (sub-AS).

Step 9  If the selection process reaches this stage, then the routes being considered are eligible for multipath task.

a  If multipath routes are being evaluated and `bgp bestpath ecmp-fast` is enabled (default) and none of the `bgp bestpath tie-break` configurations is enabled (default), then there is no preferred route in the multipath group and routes are not evaluated further. The effective ordering of routes in the multipath group is implementation-dependent and routes that were added to the multipath group first are preferred over routes that are added to the multipath group later. The path selection process ends here.

b  If multipath routes are being evaluated and `bgp bestpath ecmp-fast` is not enabled, continue to step 10.

Step 10  If the routes are eligible for multipath task and none of the `bgp bestpath tie-break` configurations is enabled (default), skip ahead to step 11. Otherwise, continue as follows:

a  If `bgp bestpath tie-break router-id` is enabled, prefer the route with the lowest ROUTER_ID. If the route is a reflected route (that is if it contains route reflector attributes), use the ORIGINATOR_ID as the ROUTER_ID for comparison.

b  If `bgp bestpath tie-break cluster-list-length` is enabled, prefer the route with the shortest CLUSTER_LIST length. The cluster list length is assumed to be 0 if the route doesn’t carry a CLUSTER_LIST attribute.

c  If there isn’t a preferred route yet, go to step 13.

Step 11  Prefer the route with the lowest ROUTER_ID. If the route is a reflected route (i.e., if it contains route reflector attributes), use the ORIGINATOR_ID as the ROUTER_ID for comparison.

Step 12  Prefer the route with the shortest CLUSTER_LIST length. The cluster list length is assumed to be 0 if the route doesn’t carry a CLUSTER_LIST attribute.

Step 13  Prefer the route received from the lowest peer address.

Step 14  If the peer address is the same for routes under comparison and `bgp additional-path receive` is enabled (the default), then the routes should differ in the received path IDs. Prefer the route with the lower received path ID. The path selection process ends here.

### 33.1.5 BGP Convergence

BGP supports convergence where it waits for all peers to join and receive all the routes from other peers. Before declaring convergence, BGP also waits for IGP protocols to converge so that all IBGP sessions are established, and routes that were learned over IBGP sessions, are resolved via the IGP routes. BGP declares convergence when it has received route updates from all its peers and EOR.
(End-Of-RIB) markers from all the expected peers and IGP protocols have converged. Using BGP convergence, you can avoid hardware updates or route advertisement churn during a switch reload or a BGP instance start.

### 33.1.6 BGP Communities

A BGP community is a group of subnet address prefixes that share a common identifying attribute. Communities simplify routing policies by consolidating IP network spaces into logical entities that BGP speakers can address to accept, prefer, and distribute routing information. BGP communities are defined by setting the community value within route maps. Community lists then reference one or more communities as follows:

- **Standard** community lists refer to communities by name or number.
- **Expanded** community lists reference communities using regular expressions.

### 33.1.7 BGP Graceful Shutdown Community

Autonomous System Boundary Routers (ASBRs) do not update all paths received from external BGP sessions and routers. They hide inefficient alternate paths and update only best paths in the routing table. BGP route policies are applied to all internal BGP sessions of ASBRs that support the graceful shutdown procedure.

As a part of maintenance mode, these route policies perform the following functionalities on routing advertisements:

- Match the graceful shutdown community with route map rules.
- Set the local preference attribute value of the paths that are tagged with the graceful shutdown community as zero.

Refer to Chapter 10: Maintenance Mode for the detailed information on maintenance mode.

### 33.1.8 BGP Labeled-Unicast (LU) path Nexthop resolution over Tunnel RIB Entries

BGP Labeled-Unicast protocol (BGP LU) path next-hop is enhanced to allow BGP in “ribd” mode to support resolution of BGP LU path next-hop over entries in the Tunnel RIB and fall-back to resolving over connected route when there is no entry in Tunnel RIB that provides a direct match for the BGP LU path next-hop. Previously, BGP in “ribd” mode allowed resolution of BGP Labeled-Unicast protocol (BGP LU) path next-hop over only connected routes, resolution of the next-hop over IGP or static routes was not allowed since the next-hop router may not be in the MPLS forwarding path in which case the traffic will get dropped by the next-hop router (per IGP).

**Note**

There are no new CLIs or show commands added with this enhancement.

The following two use cases explain how BGP LU path next-hop resolution over tunnels would help in achieving desired or efficient traffic forwarding.

- **Egress Peer Engineering (EPE)**
- **Inter-AS Option C**

**Egress Peer Engineering (EPE)**

Egress Peer Engineering is a source-routing paradigm that provides ability to select an egress node/interface through which traffic goes out of an Autonomous System (AS). As shown in Figure 1 below R1, R2, ASBR1 & ASBR2 are in AS 1 and E1, E2, E3 & E4 are in different Ases. R1, R2, ASBR1 & ASBR2 could be connected each other directly or reachable to each other over an IGP (OSPF/ISIS)
or MPLS tunnel. Let’s assume reachability of loop-back addresses 1.1.1.1, 2.2.2.2, 3.3.3.3 & 4.4.4.4 through LDP or Segment Routing (SR). There exists an iBGP Full Mesh between R1, R2, ASBR1 & ASBR2. eBGP session is present between ASBR1 & E1, ASBR1 & E2, ASBR2 & E3 and ASBR2 & E4. Consider following BGP updates are received on ASBR1:

Prefix 50.0.0.0/8 next-hop 10.0.0.2 as-path 2 100 from E1
Prefix 50.0.0.0/8 next-hop 11.0.0.2 as-path 3 200 300 from E2

BGP path from E1 will be selected as best path due to shorter AS path length. ASBR1 advertises this prefix to both R1 & R2. Any traffic destined to prefix 50.0.0.0/8 from R1 will always be tunneled to ASBR1 and then it will always be sent on an interface connected to E1. Traditional Destination based routing enforced by BGP policy and best path selection on the ASBRs may route traffic to a single AS as exit when a case can be made that for some prefixes an exit via some other AS may be preferable. BGP LU can be used here to perform traffic engineering or selecting Egress peer through which traffic should be forwarded.

A Centralized EPE Controller can be used to establish iBGP session with R1 & R2. Let’s assume Controller advertises BGP LU routes for E2, i.e., 11.0.0.2/32, with next-hop set to loop-back IP address of ASBR1, that is, 1.1.1.1 and a label 111 to R1 & R2.

Switch# show ip bgp 11.0.0.2/32
BGP routing table information for VRF default
Router identifier 3.3.3.3, local AS number 1
BGP routing table entry for 11.0.0.2/32
Paths: 1 available
Local
1.1.1.1 labels [111] from 100.100.100.1 (100.100.100.1)
Origin IGP, metric 0, localpref 100, IGP metric 40, weight 0, received
21:07:07 ago, valid, external, not installed
Rx SAFI: Labels
Tunnel RIB eligible
BGP LU path next-hop will get resolved over an ISIS SR tunnel present on R1 & R2 to reach 1.1.1.1, loop-back IP address of ASBR1.

```
Switch# show tunnel rib brief
  Endpoint   Tunnel Type Index(es)   Metric  Metric2 Preference Preference2
  ----------- ------------ ---------   ------- ------- -----------  -----------
  1.1.1.1/32  IS-IS SR IPv4     5         40       0       115           0
```

```
Switch# show bgp labeled-unicast tunnel
  Index  Endpoint     Nexthop/Tunnel Index Interface Labels Contributing Metric
  -----  -----------     -------------------- --------- ------ ------------ ------
  1     11.0.0.2/32  IS-IS SR IPv4 (5)        -     [ 111 ]    Yes        0
```

```
Metric 2 Pref Pref 2
-------- ---- ----
100     200    0
```

```
Switch# show isis segment-routing tunnel
  Index   Endpoint          Nexthop            Interface        Labels
  -----   -----------        -------            ---------        ----------
  5      1.1.1.1/32        6.6.6.6            Ethernet 5       [ 900001 ]
```

Controller or CLI can be used to install a static label route on ASBR1 such that ingress label 111 have a forwarding action of ‘POP and forward’ to next-hop (11.0.0.2) in MPLS forwarding table.

```
Switch# show mpls lfb route
  MPLS forwarding table (Label [metric] Vias) - 20 routes
  MPLS next-hop resolution allow default route: False
  Via Type Codes:
      M - Mpls Via, P - Pseudowire Via,
      I - IP Lookup Via, V - Vlan Via,
      VA - EVPN Vlan Aware Via, ES - EVPN Ethernet Segment Via,
      VF - EVPN Vlan Flood Via, AF - EVPN Vlan Aware Flood Via,
      NG - Nexthop Group Via
  Source Codes:
      S - Static MPLS Route, B2 - BGP L2 EVPN,
      B3 - BGP L3 VPN, R - RSVP,
      P - Pseudowire, L - LDP,
      IP - IS-IS SR Prefix Segment, IA - IS-IS SR Adjacency Segment,
      IL - IS-IS SR Segment to LDF, L1 - LDP to IS-IS SR Segment,
      BL - BGP LU, ST - SR TE Policy,
      DE - Debug LFIB
S   111     [100]
      via M, 11.0.0.2, pop
      payload ipv4, apply egress-acl
      interface Ethernet 4
```
For prefixes to which traffic should be sent over interface connected E2 controller will advertise a BGP route with next-hop being BGP LU prefix and higher local-preference compared to paths advertised by ASBR1 & ASBR2, so that path received from controller will be preferred over paths coming from ASBR1 & ASBR2.

Switch# show ip bgp 50.0.0.0/8
BGP routing table information for VRF default
Router identifier 3.3.3.3, local AS number 1
BGP routing table entry for 50.0.0.0/8
Paths: 3 available
Local
   11.0.0.2 from 100.100.100.1 (100.100.100.1)
     Origin IGP, metric 0, localpref 200, IGP metric 0, weight 0, received 00:00:15 ago, valid, internal, best
     Rx SAFI: Unicast
   2 100
     1.1.1.1 from 1.1.1.1 (1.1.1.1)
     Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:04:49 ago, valid, internal
     Rx SAFI: Unicast
   2 200 300
     2.2.2.2 from 2.2.2.2 (2.2.2.2)
     Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, received 00:30:38 ago, valid, internal
     Rx SAFI: Unicast

This results in pushing 2 labels on R1, top label is the label corresponding to ISIS SR tunnel to reach ASBR1 and bottom label is the label that corresponds to egress interface. Similarly LU route for 12.0.0.0.2 or 13.0.0.2 can be advertised from controller to select egress peer between E3 & E4. This approach provides Egress peer selection on an ingress router R1/R2.

Switch# show ip route 50.0.0.0/8
VRF: default
Codes: C - connected, S - static, K - kernel,
    O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
    E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
    N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
    R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
    O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
    NG - Nexthop Group Static Route, V - VXLAN Control Service,
    DH - DHCP client installed default route, M - Martian,
    DP - Dynamic Policy Route

B I   50.0.0.0/8 [200/0] via 11.0.0.2/32, BGP LU tunnel index 1
     via 6.6.6.6, Ethernet 5, label 900001 111

Inter-AS Option C

Inter-AS Option C is an efficient and scalable MPLS IP VPN solution to provide connectivity between two sites of a customer connected to Provider Edge (PE) routers in different ASes. Following diagram shows a typical topology.
PE1 & ASBR1 and PE2 & ASBR2 distribute loop-back addresses using an IBGP Labeled Unicast (LU) session. ASBR2 advertises system addresses in AS200 to ASBR1 with next-hop as itself over EBGP LU session between them and installing Label swap entry of label sent to ASBR1 (L2) to label received from PE2 (L1) in MPLS forwarding table. ASBR1 further propagates system addresses in AS200 learned from ASBR2 into AS100 or to PE1 using IBGP LU session with next-hop as itself and installing Label swap entry with label advertised to PE1 (L3) to Label received from ASBR2 (L2) in MPLS forwarding table. Similarly ASBR1 sends system addresses in AS100 to ASBR2 over EBGP LU session, ASBR2 forwards them into AS200 or to PE2 using IBGP LU session with itself as next-hop and this would trigger installing appropriate label swap actions into MPLS forwarding table. These advertisements results in the creation of a label switched path from PE1 to PE2.

PE1 & PE2 exchange VPN routes between each other using a Multi hop EBGP session with next-hop being their own loop-back/system addresses. This method eliminates the requirement of storing or sending/receiving VPN routes at ASBR routers. When PE & ASBR routers are non-adjacent, but in the same AS, then LDP or ISIS-SR can be used as a transport label signaling protocol and this would need resolving BGP LU path next-hop over LDP or ISIS-SR tunnel. An IP packet destined to an address in CE1 site 2 is received on PE1 from CE1 site 1 PE1 would need to push 3 labels onto it. Bottom label corresponds to packet destination address in a particular VRF of CE1 site 2 advertised by PE2 to PE1 over Multi hop EBGP session, Middle label belongs to PE2 system address sent by ASBR1 and top label corresponding to ASBR1 system address assigned by transport label signaling protocol.
33.2 Configuring BGP

These sections describe basic BGP configuration steps:

- Section 33.2.1: Configuring BGP Instances
- Section 33.2.2: Configuring BGP Neighbors
- Section 33.2.3: Configuring GTSM for BGP
- Section 33.2.4: Configuring Routes
- Section 33.2.5: Configuring Address Families
- Section 33.2.6: Configuring Best-path Selection
- Section 33.2.7: Configuring BGP Convergence
- Section 33.2.8: Configuring BGP Graceful Shutdown Community
- Section 33.2.9: BGP Confederations
- Section 33.2.10: BGP Operational Commands

33.2.1 Configuring BGP Instances

33.2.1.1 Creating an Instance and Entering BGP Configuration Mode

The switch supports one BGP instance, which is associated with a specified autonomous system (AS). To other BGP peers, the AS number uniquely identifies the network to which the switch belongs. Arista switches support four-byte AS numbers as described in RFC 4893. Four-byte AS number capability is communicated to BGP peers in OPEN messages. When communicating with a BGP peer which does not support four-byte AS numbers, the switch will replace AS numbers greater than 65535 with the well-known two-byte AS number 23456 (also called AS_TRANS), and encode the actual four-byte AS numbers using the AS4_PATH attribute.

The switch must be in router-BGP configuration mode to run BGP configuration commands. The `router bgp` command places the switch in router-BGP configuration mode for creating a BGP instance if one was not previously created. BGP configuration commands apply globally to the BGP instance.

**Example**

- This command places the switch in router-BGP configuration mode. It also creates a BGP instance in AS 50 if an instance was not previously created.

  switch(config)#router bgp 50
  switch(config-router-bgp)#

When a BGP instance exists, the `router bgp` command must include its autonomous system. Any attempt to create a second instance results in an error message.

**Example**

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

  switch(config)#router bgp 100
  % BGP is already running with AS number 50
  switch(config)#
### Chapter 33: Border Gateway Protocol (BGP)

#### Configuring BGP

**33.2.2 Configuring BGP Neighbors**

**33.2.2.1 Establishing BGP Neighbors**

BGP neighbors, or peers, are established by configuration commands that initiate a TCP connection. BGP supports two types of neighbors:

- Internal neighbors are in the same autonomous system.
- External neighbors are in different autonomous systems.

BGP neighbors can be either static or dynamic:

- Static neighbors are established by manually configuring the connection.
- Dynamic neighbors are established by creating a listen range and accepting incoming connections from neighbors in that address range.

Static neighbors may belong to a static peer group, allowing them to be configured as a group. Configuration applied to an individual member of a static peer group overrides the group configuration for that peer. Dynamic neighbors must belong to a dynamic peer group, and can only be configured as a group.

**Static BGP Neighbors**

The `neighbor remote-as` command connects the switch with a peer, establishing a static neighbor.

Once established, a static neighbor may be added to an existing peer group. Any configuration applied to the peer group then is inherited by the neighbor, unless a conflicting configuration has been entered for that peer. Settings applied to a member of the peer group override group settings.
To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (for example, in an IPv6-only environment), configure an appropriate router ID using the `router-id` (BGP) command.

**Example**

- These commands establish an internal BGP connection with the peer at 10.1.1.14.

```plaintext
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 10.1.1.14 remote-as 50
```

- These commands establish an external BGP connection with the peer at 192.168.2.5.

```plaintext
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 192.168.2.5 remote-as 100
```

**Dynamic BGP Neighbors**

The `bgp listen range` command specifies a range of IPv4 addresses from which the switch will accept incoming dynamic BGP peering requests, and creates the named dynamic peer group to which those peers belong. Dynamic BGP neighbors are peers which have not been manually established, but are accepted into a dynamic peer group when the switch receives a peering request from them.

Dynamic peers cannot be configured individually, but inherit any configuration that is applied to the peer group to which they belong. Peering relationships with dynamic peers are terminated if the peer group is deleted.

**Example**

- These commands create a peer group called “brazil” which accepts dynamic peering requests from the 192.168.2.0/24 subnet.

```plaintext
switch(config)#router bgp 50
switch(config-router-bgp)#bgp listen range 192.168.2.0/24 peer-group brazil
remote-as 50
```

**Displaying Neighbor Connections**

The `show ip bgp summary` and `show ip bgp neighbors` commands display neighbor connection status.

**Example**

- This command indicates the connection state with the peer at 192.168.2.5 is `Estab` (established). The peer is an external neighbor because it is in AS 100 and the local server is in AS 50.

```plaintext
switch>show ip bgp summary
BGP summary information for VRF default
BGP router identifier 192.168.104.2, local AS number 50
Neighbor Status Codes: m - Under maintenance
   Neighbor V AS  MsgRcvd  MsgSent  InQ OutQ   Up/Down State  PfxRcd PfxAcc
192.168.2.5 4  100  198  281 0 0 03:11:31 Estab 12 12
```

2198
Static BGP Peer Groups

A static BGP peer group is a collection of BGP neighbors which can be configured as a group. Once a static peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Neighbors added to the group will inherit any settings already created for the group. Static peer group members may also be configured individually, and the settings of an individual neighbor in the peer group override group settings for that neighbor.

When the `default` form of a BGP configuration command is entered for a member of a static peer group, the peer inherits that configuration from the peer group.

A static peer group is created with the `neighbor peer group (create)` command, or by using the `bgp listen range` command to accept dynamic peering requests. Once a static peer group has been created, static neighbors can be manually added to the group by using the `neighbor peer-group (neighbor assignment)` command. The `no neighbor peer-group (neighbor assignment)` command removes a neighbor from a static peer group.

The `no neighbor peer group (create)` command will delete a static peer group. When a peer group is deleted, the members of that group revert to their individual configurations, or to the system default for any attributes that have not been specifically configured for that peer.

Examples

- These commands create a peer group named “akron.”

  ```
  switch(config)#router bgp 50
  switch(config-router-bgp)#neighbor akron peer-group
  switch(config-router-bgp)#
  ```

- This command adds the neighbors at 1.1.1.1 and 2.2.2.2 to peer group akron.

  ```
  switch(config-router-bgp)#neighbor 1.1.1.1 peer-group akron
  switch(config-router-bgp)#neighbor 2.2.2.2 peer-group akron
  switch(config-router-bgp)#
  ```

- These commands configure the members of peer group akron, but cause the neighbor at 1.1.1.1 to use the system default value for out-delay.

  ```
  switch(config-router-bgp)#neighbor akron remote-as 109
  switch(config-router-bgp)#neighbor akron out-delay 101
  switch(config-router-bgp)#neighbor akron maximum-routes 12000
  switch(config-router-bgp)#no neighbor 1.1.1.1 out-delay
  switch(config-router-bgp)#
  ```

Dynamic BGP Peer Groups

A dynamic BGP peer group is a collection of BGP neighbors in a specified address range which makes a peer request to the switch. Members of dynamic peer group are configured in groups and not as individuals. A dynamic peer group name is used as a parameter to apply the configuration across all the members in the group. Neighbors joining the group inherit any settings already created for the group.

The `bgp listen range` command is used to create a dynamic peer group. This command identifies the BGP peering request from a range of IP address, and names the dynamic peer group to which those peers belong to. The `bgp listen range` command can be configured to accept a peering request from a single AS number or to accept peer request from the range of AS numbers. To accept the request from the range of AS numbers use the `peer filter` option in the command as shown. If the peer filter referred by the `bgp listen range` command does not exist, or if the filter exists but has no match commands, it will accept any AS number.
When a listen range command is modified, any existing dynamic neighbor that is already established will get reset.

To delete a dynamic peer group, use the no or default form of the `bgp listen range` command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

**Example**

- These commands create a dynamic peer group called “brazil” in a single AS, which accepts peering requests from the 192.0.2.0/24 subnet the single AS is 5.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 192.0.2.0/24 peer-group brazil
remote-as 5
```

- These commands create a dynamic peer group called “brazil” in a range of ASNs, which accepts peering requests from the 192.0.2.0/24 subnet. The range of AS numbers is defined by peer filter option.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 192.0.2.0/24 peer-group brazil
peer-filter group-1
```

The `show ip bgp peer-group` command displays the source of a listen range’s remote AS number definition as shown.

```
switch(config-router-bgp)#show ip bgp peer-group
BGP peer-group is brazil
BGP version 4
Listen-range subnets:
VRF default:
  192.0.2.0/24, remote AS 5
  192.0.2.0/24, peer filter group1
```

**33.2.2.2 Peer Filter**

A peer filter defines a set of rules which decides whether to accept or reject the incoming peer request, based on the specific peer attributes. The peer filter is defined using a sequence number and a match statement, today it does support one new match statement for matching against a range of BGP AS numbers. A peer filter is defined using a peer filter configuration mode as shown. The peer filter command supports only matching AS ranges. Unlike route maps, peer filter do not support sets, continues or subroutines.

To delete a peer filter, use the no or default form of the peer filter command.

**Example**

- These commands define a peer filter that accepts any AS number.

```
switch(config)#peer-filter group1
switch(config-peer-filter-group1)#10 match as-range 1-4294967295 result accept
```

- These commands define a peer filter that accepts any AS number within 65000 and 65100 (inclusive) except 65008 and 65009.

```
switch(config)#peer-filter group2
switch(config-peer-filter-group2)#10 match as-range 65008-65009 result reject
switch(config-peer-filter-group2)#20 match as-range 65000-651000 result accept
```
These commands define a peer filter that accepts 3 specific remote AS numbers.

```plaintext
switch(config)#peer-filter group3
switch(config-peer-filter-group3)#10 match as-range 65003 result accept
switch(config-peer-filter-group3)#20 match as-range 65007 result accept
switch(config-peer-filter-group3)#30 match as-range 65009 result accept
```

The `show peer-filter` command displays the peer filter definition.

```plaintext
switch(config)#show ip bgp peer-group3
peer-filter group3
  10 match as-range 65003 result accept
  20 match as-range 65007 result accept
  30 match as-range 65009 result accept
```

33.2.2.3 Special Considerations for IPv6

BGP predates the use of IPv6, and BGP configuration assumes IPv4 connections by default. The following additional steps are used to configure IPv6 BGP neighbors.

**Note**

To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the `router-id (BGP)` command.

**Activating IPv6 Neighbors**

By default, the switch does not negotiate or advertise IPv6 BGP routes. In order to establish a session with an IPv6 neighbor, it must be made active in the IPv6 address family. The `bgp default ipv6-unicast` command causes the switch to send IPv6 capability messages and all network advertisements with IPv6 prefixes to all BGP neighbors. The `neighbor activate` command issued in IPv6 address family configuration mode does the same for a single BGP neighbor.

**Examples**

- These commands make all BGP neighbors active in the IPv6 address family.
  ```plaintext
  switch(config)#router bgp 11
  switch(config)#address-family ipv6
  switch(config-router-bgp-af)#bgp default ipv6-unicast
  switch(config-router-bgp-af)#exit
  switch(config-router-bgp-af)#
  ```

- These commands make the BGP neighbor at 2001:0DB8:8c01::1 active in the IPv6 address family.
  ```plaintext
  switch(config)#router bgp 11
  switch(config)#address-family ipv6
  switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
  switch(config-router-bgp-af)#exit
  switch(config-router-bgp-af)#
  ```

**Sending IPv4 NLRIs over IPv6 Connections**

The switch supports the exchange of IPv4 NLRIs with IPv6 neighbors. To enable this feature for all IPv6 neighbors, use the command `bgp default ipv4-unicast transport ipv6` in IPv4 address family configuration mode. To enable it for a single IPv6 neighbor, use the `neighbor activate` command for that neighbor in IPv4 address family configuration mode.
To send IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors, then configure the switch to automatically select a local IPv4 address to be sent in NLRI transport to the IPv6 neighbors in a peer group called “indianapolis”.

```bash
switch(config)#router bgp 11
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp default ipv4-unicast transport ipv6
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#neighbor indianapolis auto-local-addr
```

These commands permit IPv4 NLRI transport with the IPv6 neighbor at 2001:0DB8:8c01::1 using a local IPv4 address of 10.7.5.11.

```bash
switch(config)#router bgp 11
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 local-v4-addr 10.7.5.11
```

**33.2.2.4 Maintaining Neighbor Connections**

BGP neighbors maintain connections by exchanging keepalive, UPDATE, and NOTIFICATION messages. Neighbors that do not receive a message from a peer within a specified period (hold time) close the BGP session with that peer. Hold time is typically three times the period between scheduled keepalive messages. The default keepalive period is 60 seconds; default hold time is 180 seconds.

The `timers bgp` command configures the hold time and keepalive period. A peer retains its BGP connections indefinitely when its hold time is zero.

**Example**

This command sets the keepalive period to 15 seconds and the hold time to 45 seconds.

```bash
switch(config-router-bgp)#timers bgp 15 45
```

The `show ip bgp neighbors` command displays the hold time.
Example

- This command indicates the BGP hold time is 45 seconds.

```bash
switch> show ip bgp neighbors 10.100.100.2
BGP neighbor is 10.100.100.2, remote AS 100
BGP version 4, remote router ID 192.168.100.13, VRF default
Negotiated BGP version 4
Last read 00:00:05, last write 00:00:05
Hold time is 45, keepalive interval is 15 seconds
Configured hold time is 45, keepalive interval is 15 seconds
Connect timer is inactive
Idle-restart timer is inactive
BGP state is Established, up for 04:44:05
Number of transitions to established: 11
Last state was OpenConfirm
Last event was RecvKeepAlive
Last sent notification: Cease/administrative reset, Last time 04:44:09
Last rcvd notification: Cease/peer de-configured, Last time 2d02h, First time 7d08h, Repeats 1
Neighbor Capabilities:
  Multiprotocol IPv4 Unicast: advertised and received and negotiated
  Four Octet ASN: advertised and received
```

33.2.2.5 Neighbor – Route Configuration

Maximum Routes

The `neighbor maximum-routes` command determines the number of BGP routes the switch accepts from a specified neighbor. The switch disables peering with the neighbor when this number is exceeded.

Example

- This command configures the switch to accept 15,000 routes from the peer at 192.168.18.24.

```bash
switch(config-router-bgp)# neighbor 192.168.18.24 maximum-routes 15000
switch(config-router-bgp)#
```

Route Reflection

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers; they do not re-advertise iBGP-learned routes within the AS to prevent routing loops. Although a fully meshed network topology ensures that all AS members share routing information, this topology can result in high volumes of iBGP messages when scaled. Alternatively, one or more routers are configured as route reflectors in larger networks.

A route reflector re-advertises routes learned through iBGP to a group of BGP neighbors within the AS, replacing the function of a fully meshed topology. The `neighbor route-reflector-client` command configures the switch to act as a route reflector and configures the specified neighbor as a client. The `bgp client-to-client reflection` command enables client-to-client reflection.

When using route reflectors, an AS is divided into clusters. A cluster contains at least one route reflector and a group of clients to which they re-advertise route information. A cluster may contain multiple route reflectors to provide redundancy protection. Each reflector has a cluster ID. When the cluster has a single route reflector, the cluster ID is its router ID. When a cluster has multiple route
reflectors, a 4-byte cluster ID is assigned to all route reflectors in the cluster, allowing them to recognize updates from other cluster reflectors. The `bgp cluster-id` command configures the cluster ID in a cluster with multiple route reflectors.

**Example**

- These commands configure the switch as a route reflector and the neighbor at 172.72.14.5 as one of its clients, and set the cluster ID to 172.22.30.101.

  ```
  switch(config-router-bgp)#neighbor 172.72.14.5 route-reflector-client
  switch(config-router-bgp)#bgp cluster-id 172.22.30.101
  switch(config-router-bgp)#
  ```

Usually the clients of a route reflector are not interconnected, and any routes learned by a client are mirrored to other clients and re-advertised within the AS by the route reflector. If the clients of a route reflector are fully meshed, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled (`no bgp client-to-client reflection`).

**Route Preference**

The primary function of external peers is to distribute routes they learn from their peers. Internal peers receive route updates without distributing them. External peers receive route updates, then distribute them to internal and external peers.

*Local preference* is a metric that iBGP sessions use to select an external route. Preferred routes have the highest local preference value. UPDATE packets include this metric in the LOCAL_PREF field.

The `neighbor export-localpref` command specifies the LOCAL_PREF that the switch sends to an internal peer. The command overrides previously assigned preferences and has no effect on external peers.

**Example**

- This command configures the switch to enter 200 in the LOCAL_PREF field of UPDATE packets it sends to the peer at 10.1.1.45.

  ```
  switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
  switch(config-router-bgp)#
  ```

The `neighbor import-localpref` command assigns a local preference to routes received through UPDATE packets from an external peer. This command has no effect when the neighbor is an internal peer.

**Example**

- This command configures the switch to assign the local preference of 50 for routes advertised from the peer at 172.16.5.2.

  ```
  switch(config-router-bgp)#neighbor 172.16.5.2 import-localpref 50
  switch(config-router-bgp)#
  ```

The `show ip bgp` command displays the LOCAL_PREF value for all listed routes.
Example

- This command indicates the route to network 10.10.20.0/24 has a local preference of 400.

```
switch# show ip bgp
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt;Ec 10.10.20.0/24</td>
<td>192.168.31.3</td>
<td>0</td>
<td>400</td>
<td>0</td>
<td>64521 i</td>
</tr>
</tbody>
</table>
```

Graceful Restart

Graceful BGP restart allows a BGP speaker with separate control plane and data plane processing to continue forwarding traffic during a BGP restart. Its neighbors (receiving speakers) may retain routing information from the restarting speaker while a BGP session with it is being re-established, reducing route flapping.

Arista switches can act as helpers (receiving speakers) for graceful BGP restart with neighbors that advertise graceful restart capability.

Graceful restart helper mode is enabled by default, but can be turned off globally with the `no graceful-restart-helper` command. Per-peer configuration takes precedence over the global configuration.

Example

- This command disables graceful restart helper mode for all BGP peers.

```
switch(config-router-bgp)# no graceful-restart-helper
switch(config-router-bgp)#
```

- This command disables graceful restart helper mode for the neighbor at 192.168.32.5 regardless of global configuration.

```
switch(config-router-bgp)# no neighbor 192.168.32.5 graceful-restart-helper
switch(config-router-bgp)#
```

Peers with graceful restart capability advertise a restart time value as an estimate of the time it will take them to restart a BGP session. When a BGP session with a restarting speaker goes down, the switch (receiving speaker) marks routes from that peer as stale and starts the restart timer. If the session with the peer is not re-established before the restart time runs out, the switch deletes the stale routes from that peer. If the session is re-established within that time, the stale path timer is started. If the stale paths are not updated by the restarting speaker before the stale path time runs out, they are deleted. The maximum time these stale paths will be retained after the BGP session is re-established is 300 seconds by default, but can be configured using the `graceful-restart stalepath-time` command.

Example

- This command configures BGP to discard stale paths from a restarting peer 500 seconds after the BGP session with that peer is re-established.

```
switch(config-router-bgp)# graceful-restart stalepath-time 500
switch(config-router-bgp)#
```
33.2.2.6 Filtering Routes

Filtering with Route Maps

Route maps are used in BGP to directly filter IPv4 unicast routes. The neighbor route-map (BGP) command applies a route map to inbound or outbound BGP routes. To display the route maps associated with a specific BGP neighbor, use the show ip bgp neighbors command.

Filtering with BGP Communities

Community values are assigned to a set of subnet prefixes through route map set commands. Route map match commands subsequently use community values to filter routes. The switch uses the following ip community-list commands to filter community routes into a BGP domain:

- `ip community-list` creates a community list by explicitly referencing one or more communities by name or number.
- `ip community-list regexp` creates a community list by referencing one or more communities by regular expression.
- `ip extcommunity-list` creates an extended community list to identify routes for VRFs or for link bandwidth (LBW) by explicitly referencing extended communities by prefix and number.
- `ip extcommunity-list regexp` creates an extended community list to identify routes for VRFs or for link bandwidth (LBW) by regular expression.

The BGP community attribute is a 32 bit value formatted as follows:

- an integer between 0 and 4294967040.
- AA:NN, where AA is 65535 and NN specifies the community number (0-65535) within the AS.

These four community attribute values, and the associated BGP speaker actions, are predefined:

- `no-export`: speaker does not advertise the routes beyond the BGP domain.
- `no-advertise`: speaker does not advertise the routes to any BGP peers.
- `local-as`: speaker does not advertise route to any external peers.
- `internet`: speaker advertises the route to the Internet community. By default, this includes all prefixes.

Example

- These commands assign two network subnets to a prefix list, assign a community number to the prefix list members, then utilize that community in an ip community-list command to permit the routes into the BGP domain.

  **Step 1** Compose the IP prefix list.

  ```
  switch(config)#ip prefix-list PL_1 permit 10.1.2.5/24
  switch(config)#ip prefix-list PL_1 permit 10.2.5.1/28
  switch(config)#
  ```

  **Step 2** Create a route map that matches the IP prefix list and sets the community value.

  ```
  switch(config)#route-map MAP_1 permit
  switch(config-route-map-MAP_1)#match ip address prefix-list PL_1
  switch(config-route-map-MAP_1)#set community 500
  switch(config-route-map-MAP_1)#exit
  ```

  **Step 3** Create a community list that references the community.

  ```
  switch(config)#ip community-list CL_1 permit 500
  switch(config)#
  ```
BGP extended communities identify routes for VRFs or for link bandwidth (LBW). Extended community clauses utilize route target (rt) and site of origin options (soo):

- **route targets** identify sites that may receive appropriately tagged routes.
- **site of origin** identifies the site where the router learned the route.

### Filtering with AS Path Access Lists

An AS path access list is a named list of permit and deny statements which use regular expressions to filter BGP routes based on their AS path attribute. AS path access lists are created using the `ip as-path access-list` command, and are applied using a route map `match` clause with the name of the access list as a parameter.

**Example**

- These commands create an AS path access list identifying routes which pass through AS 3, create a route map which references the access list, assign the routes it filters to community 300, and apply the route map to the neighbor at 192.68.14.5 to assign a community value of 300 to inbound routes received from that neighbor.

**Step 1** Create the AS path access list.

```bash
switch(config)#ip as-path access-list as_list3 permit _3_
```

**Step 2** Create a route map that matches the AS path access list and sets the community value.

```bash
switch(config)#route-map MAP_3 permit
switch(config-route-map-MAP_3)#match as-path as_list3
switch(config-route-map-MAP_3)#set community 300
switch(config-route-map-MAP_3)#exit
```

**Step 3** Apply the route map to the neighbor.

```bash
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.68.14.5 route-map MAP_3 in
```

### Configuring GTSM for BGP

The Generalized TTL Security Mechanism (GTSM) uses a packet's Time to Live (TTL) (IPv4) or Hop Limit (IPv6) to protect BGP peering sessions from denial-of-service (DoS) attacks based on forged protocol packets.

An IP packet received from a BGP peer is discarded when its current TTL value is less than \((255-n)\) where \(n\) is the configured maximum number of hops to the peer. Use the `neighbor ttl maximum-hops` command to configure the maximum hop count.

**Note** IP packets to GTSM enabled BGP peers are sent with the configured TTL value of 255.

### Configuring Routes

#### Advertising Routes

A BGP neighbor advertises routes it can reach through UPDATE packets. The `network (BGP)` command specifies a prefix that the switch advertises as a route originating from its AS.

The configuration clears the host portion of addresses entered in `network` commands. For example, 192.0.2.4/24 is stored as 192.0.2.0/24.
Example

- This command configures the switch to advertise the 10.5.8.0/24 network.

```
switch(config-router-bgp)#network 10.5.8.0/24
switch(config-router-bgp)#
```

By default, BGP will advertise only those routes that are active in the switch’s RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The `bgp advertise-inactive` command causes BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

Examples

- This command configures the switch to advertise routes learned through BGP even if they are not active on the switch.

```
switch(config-router-bgp)#bgp advertise-inactive
switch(config-router-bgp)#
```

- This command overrides inactive route advertisement for VRF “purple.”

```
switch(config-router-bgp)#vrf purple
switch(config-router-bgp-vrf-purple)#no bgp advertise-inactive
switch(config-router-bgp-vrf-purple)#
```

33.2.4.2 Advertising ISIS Routes into BGP Network

The `redistribute isis route-map isis-to-bgp` command advertises the routes learned through IS-IS routes into BGP network. It also allows to selectively advertise some routes and modify route attributes before advertising using route-maps.

The command is available in both address-family mode and router BGP mode, however, the command is rejected if configured in both address-family mode and router mode at the same time.

While redistributing IS-IS routes into BGP, Level-1|Level-2 keyword can be used to selectively redistribute Level-1 routes or Level-2 routes into BGP. The `level-1|level-2` keyword is optional and it defaults to level-2 when not configured.

Use `show ipv6 bgp <detail>` command to verify that routes are advertised with correct attributes.

Note

If the command is configured in router-af mode, it only redistributes routes with matching address family. If it is configured in router mode, it applies to all enabled address-families.

Examples

- In this example the `redistribute isis route-map isis-to-bgp` command redistributes the IS-IS routes into BGP, in `address-family` mode.

```
Switch(config)#router bgp 1
Switch(config-router-bgp)#address-family ipv4
Switch(config-router-bgp-af)#redistribute isis level-1 route-map isis-to-bgp-v4
```

- In this example the `redistribute isis route-map isis-to-bgp` command redistributes the IS-IS routes into BGP, in `router bgp` mode.

```
Switch(config)#router bgp 1
Switch(config-router-bgp)#redistribute isis level-1 route-map isis-to-bgp
```
33.2.4.3 BGP Route Aggregation

Aggregation combines the characteristics of multiple routes into a single route for advertisement by the BGP speaker. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregation options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Aggregate routes are created with the `aggregate-address` command, which takes an IP subnet as an argument; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.

BGP speakers display aggregate routes that they create as null routes (with one exception: if all the contributors to the aggregate have the same BGP path attributes, then the BGP aggregate copies those attributes and is no longer a null route). Aggregate routes are advertised into the BGP autonomous system and redistributed automatically, and their redistribution cannot be disabled. BGP neighbors display inbound aggregate routes as normal BGP routes. Null routes are displayed with the `show ip route` command; normal BGP routes (and null aggregate routes) are displayed with the `show ip bgp` and `show ip route` commands.

Aggregation Options

The `aggregate-address` command provides the following aggregate route options:

- **AS_PATH attribute inclusion:** the `as-set` option controls the aggregate route’s AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

  When the command includes `as-set`, the aggregate route’s AS_SET attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the AS_SET.

  When the command does not include `as-set`, the aggregate route’s ATOMIC_AGGREGATE attribute is set and the AS_PATH attribute does not include AS numbers of contributing routes.

- **Attribute assignment:** The `attribute-map` option assigns attributes contained in set commands in a specified route map’s lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route’s attributes by the switch.

- **Route suppression:** The `summary-only` option suppresses the advertisement of the contributor routes that comprise the aggregate.

- **Contributor filtering:** The `match-map` option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

Example

- These commands create an aggregate route (10.16.48.0/20) from four contributor routes (10.16.48.0/23, 10.16.50.0/23, 10.16.52.0/23, and 10.16.54.0/23). The aggregate route includes the AS_PATH information from the contributor routes.

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#aggregate-address 10.16.48.0/20 as-set
  switch(config-router-bgp)#exit
  switch(config)#
  ```
These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)#route-map map1 permit 10
switch(config-route-map-map1)#set local-preference 40
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 10.16.48.0/20 attribute-map map1
```

These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

```
switch(config)#route-map matchmap permit 10
switch(config-route-map-matchmap)#match ip address prefix-list agglist
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 1.1.0.0/16 match-map matchmap
```

### 33.2.4.4 Customizing the BGP AS-Path Attribute

The BGP Replace AS-Path feature enables customizing the AS_PATH attribute for prefixes that are either received from a BGP neighbor or advertised to a BGP neighbor. To configure the BGP Replace AS-Path feature, use the `set as-path match` and `set as-path prepend` commands.

To replace the `AS_PATH` attribute of routes received from a BGP neighbor, configure a route map and attach the policy to the corresponding BGP neighbor statement in the inbound direction.

To replace the `AS_PATH` attribute of routes that are advertised to a neighbor, configure a route map and attach the policy to the corresponding BGP neighbor statement in the outbound direction.

The Replace AS-Path feature works in conjunction with the AS-Path Prepend feature which is also used to modify the AS_PATH attribute. However, if both features are configured within the same route map, then the replace AS-Path feature takes precedence over the AS-Path Prepend.

#### Note

The BGP Replace AS-Path feature supports both eBGP and iBGP neighbors. The locally configured AS number is always prefixed to the AS-Path of routes advertised to the eBGP neighbors. This RFC behavior is retained in Arista’s implementation of the Replace AS-Path feature as well.

BGP Replace AS-Path has the following limitations:

- Replacing the AS-Path should be used cautiously since it may impact BGP loop prevention.
- A few duplicated routes may be advertised and installed on a router after the original AS-Path of those routes are replaced. To fix this issue, it is always suggested to filter out such routes by prefix with BGP Community.
Example

- This command replaces the AS-Path with the “none” option.

```plaintext
switch#show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q - Queued for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 302 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 i
switch#configuration terminal
switch(config)#route-map foo permit 10
switch(config-route-map-foo)#set as-path match all replacement none
switch(config-route-map-foo)#exit
switch(config)#router bgp 200
switch(config-router-bgp)#neighbor 80.80.1.2 route-map foo out
switch(config-router-bgp)#end

switch#show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q - Queued for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

Network Next Hop Metric LocPref Weight Path
* > 101.101.1.0/24 80.80.1.1 - - - 200 i
* > 102.102.1.0/24 80.80.1.1 - - - 200 i
* > 103.103.1.0/24 80.80.1.1 - - - 200 i
* > 202.202.1.0/24 80.80.1.1 - - - 200 i
```

The AS-Path of matching prefixes are replaced with an empty or a null AS-Path. AS 302 is removed from prefix 103.103.1.0/24 as shown in the above output.
This command replaces the AS-Path with the “auto” option.

```
switch(config)#route-map foo permit 10
switch(config-route-map-foo)#set as-path match all replacement auto
switch(config-route-map-foo)#end
```

```
switch#show ip bgp neighbors 80.80.1.2 advertised-routes
BGP routing table information for VRF default
Router identifier 202.202.1.1, local AS number 200
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast, q - Queued for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop
```

```
<table>
<thead>
<tr>
<th>Network Next Hop Metric LocPref Weight Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 101.101.1.0/24 80.80.1.1 - - - 200 200 i</td>
</tr>
<tr>
<td>* &gt; 102.102.1.0/24 80.80.1.1 - - - 200 200 i</td>
</tr>
<tr>
<td>* &gt; 103.103.1.0/24 80.80.1.1 - - - 200 200 i</td>
</tr>
<tr>
<td>* &gt; 202.202.1.0/24 80.80.1.1 - - - 200 200 i</td>
</tr>
</tbody>
</table>
```

The AS-Path of matching prefixes are replaced with the locally configured AS – 200.

### 33.2.5 Configuring Address Families

The switch determines the network prefixes that peering sessions advertise and the BGP neighbor addresses that receive advertisements through address family activity configuration.

An address family is a data structure that defines route advertising status to BGP neighbor addresses. Each BGP neighbor address is assigned an activity level for each address family on the switch. The switch sends capability and network prefix advertisements to neighbor addresses that are active within specified address families:

- IPv4 address family: switch advertises IPv4 capability and network commands with IPv4 prefixes to neighbor addresses configured as **IPv4 address family active**.
- IPv6 address family: switch advertises IPv6 capability and network commands with IPv6 prefixes to neighbor addresses configured as **IPv6 address family active**.

#### 33.2.5.1 Neighbor Address Family Configuration

Address family activity levels for neighbor addresses are configured through **bgp default** and **neighbor activate** commands.

- The **bgp default** command specifies the default activity level of BGP neighbor addresses for a specified address family.
- The **neighbor activate** command specifies deviations from default address family activity level for a specified BGP neighbor address.

### Default neighbor activation

The **bgp default** command configures the default address family activity level of all configured BGP neighbor addresses. The switch advertises the following to **address family active** addresses:

- IPv4 address family active: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family active: IPv6 capability and all network advertisements with IPv6 prefixes.

These commands configure default address family activity levels for configured BGP neighbor addresses:
• `bgp default ipv4-unicast`  all BGP neighbor addresses are IPv4 address family active (this is the switch default).
• `no bgp default ipv4-unicast`  no BGP neighbor addresses are IPv4 address family active.
• `bgp default ipv6-unicast`  all BGP neighbor addresses are IPv6 address family active.
• `no bgp default ipv6-unicast`  no BGP neighbor addresses are IPv6 address family active (this is the switch default).
• `bgp default ipv4-unicast transport ipv6`  all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.

**Note**
If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command `bgp default ipv4-unicast transport ipv6`. For an individual neighbor, use the `neighbor activate` command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

**Activating Individual Neighbor Addresses**
The `address-family` command places the switch in address family mode to configure the address family activity level of individual BGP neighbor addresses. The switch supports these address families:
• `ipv4-unicast`
• `ipv6-unicast`

Running-config displays address family commands in sub-blocks of the BGP configuration. The `neighbor activate` command is available in each address family configuration mode and defines the configuration mode address family activity level of a specified configured BGP neighbor address. Addresses are assigned one of the following states by the activate command:
• `neighbor activate`  configures the address as active in the configuration mode address family.
• `no neighbor activate`  configures the address as not active in the configuration mode address family.

The switch sends the following announcements to addresses that are active in an address family:
• IPv4 address family: IPv4 capability and all network routes with IPv4 prefixes.
• IPv6 address family: IPv6 capability and all network routes with IPv6 prefixes.

The `neighbor route-map (BGP)` command applies a route map to inbound or outbound BGP routes. In address-family mode, the route map is applied to routes corresponding to the configuration mode address family. When a route map is applied to outbound routes, the switch advertises only routes matching at least one section of the route map. One outbound and one inbound route map can be applied to a neighbor for each address family. Applying a route map to a route replaces the previous corresponding route map assignment.

**Network Route Advertising in Address Families**
The `network (BGP)` command specifies a network for advertisement through UPDATE packets to BGP peers. The command is available in Router-BGP and Router-BGP-Address-Family configuration modes; the mode in which the command is issued does not affect the command's execution.
• Commands with an IPv4 address are advertised to peers that are IPv4 address family-active.
• Commands with an IPv6 address are advertised to peers that are IPv6 address family-active.

**Examples**
• These commands instantiate BGP, configure three neighbors, and configure 2 network routes.
The default activity level for IPv4 and IPv6 address families is set to the default; all neighbor addresses are IPv4 address family active and IPv6 address family not active. IPv4 capability and network routes with IPv4 prefixes are advertised to all neighbor IPv4 addresses.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands instantiate BGP on the switch, set IPv4 default activity level (not active), set IPv6 default activity level (active), and configure three neighbor addresses and two network route prefixes.

IPv6 capability and network routes with IPv6 prefixes are advertised to all neighbor addresses.

```
switch(config)#router bgp 10
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands configure three neighbors, two network routes, and the default activity level for each address family (not active), and specify neighbor addresses for each address family that is active.

```
switch(config)#router bgp 11
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#no bgp default ipv6-unicast
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#neighbor 172.21.14.8 activate
switch(config-router-bgp-af)#neighbor 172.23.18.6 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#address-family ipv6
switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

- These commands permit IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors.
33.2.6 Configuring Best-path Selection

The best-path selection algorithm (described under Best-Path Selection) determines which of multiple paths to the same destination received by BGP will be added to the IP routing table. To shape route preferences and influence best-path selection, use the following commands in router-BGP configuration mode.

- `bgp always-compare-med` configures the switch to always consider the multi-exit discriminator (MED) value when comparing paths (disabled by default).
- `bgp bestpath as-path ignore` configures the switch to ignore the length of the autonomous system (AS) path when comparing routes (disabled by default).
- `bgp bestpath as-path multipath-relax` used in equal-cost multi path (ECMP configuration) and enabled by default; the `no` form of the command configures the switch to consider paths unequal if their AS paths have different contents.
- `bgp bestpath ecmp-fast` the `no` form of this command causes the switch to ignore order of arrival in evaluating paths within an ECMP group.
- `bgp bestpath med confed` causes comparison of multi-exit discriminator (MED) values in routes originating within the same confederation as the switch and received from confederation peers (disabled by default).
- `bgp bestpath med missing-as-worst` configures the switch to treat a missing MED as having the highest (least preferred) value (disabled by default). This command overrides the `missing-as-worst` setting of the `bgp bestpath med confed` command.
- `bgp bestpath tie-break cluster-list-length` configures the switch to prefer the multipath route with the shortest CLUSTER_LIST length in case of a tie in step 10 of the selection process (disabled by default).
- `bgp bestpath tie-break router-id` configures the switch to prefer the multipath route with the lowest ROUTER_ID in case of a tie in step 10 (disabled by default).

33.2.6.1 Displaying Reasons for Best-path Selection

To see the reasons why certain routes were excluded by the best-path selection process, use the `detail` option of the `show ip bgp` command. Enter the prefix to which BGP has selected a best path, and the output will display all learned paths. Paths which were not selected as best will display the reason they were not selected after the label `not best`.

The reason will be listed as one of the following:

- Path weight
- Local preference
- AS path length
- Origin
- Path MED
- eBGP path preferred
- IGP cost
- AS path details
- ECMP-Fast configured
- Router ID
- Originator ID
- Router ID tie-break configured
• Cluster list length
• Cluster list length tie-break configured
• Peer IP address
• Path ID
• Redistributed route exists
• Unknown
• Another route from the same AS is a better BGP route
• Peer not ready
• Unusable

Example
• This command displays the reasons why three routes to 172.16.0.0/24 were rejected by the best-path algorithm. The reason for rejection is preceded by the label Not best:

```
switch> # show ip bgp 172.16.0.0/24 detail
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status: [a.b.c.d] - Route is queued for advertisement to peer.
BGP routing table entry for 204.1.47.220/30
Paths: 4 available
  64512 64550 65100
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid, external, ECMP head, ECMP, best, ECMP contributor
      Rx SAFI: Unicast
    192.168.24.2 from 192.168.24.2 (192.168.100.22)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: ECMP-Fast configured
  64512 64550 65100
    192.168.34.2 from 192.168.34.2 (192.168.100.23)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: Redistributed route exists
  64512 64550 65100
    192.168.44.2 from 192.168.44.2 (192.168.100.24)
      Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid, external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: eBGP path preferred
 Not advertised to any peer
switch>
```

33.2.7 Configuring BGP Convergence

To avoid hardware updates and route advertisement churn during switch reload or BGP instance start, BGP enters into the convergence state where it waits for all peers to join and receive all routes from all the peers.

**BGP Convergence** is bound by an upper value of convergence time (default value is 5 minutes) and BGP declares convergence on expiry of convergence timer. At the end of convergence, BGP updates the routes in FIB and advertises to all the peers.
To configure BGP convergence and the different timeout features, use the following commands in the router-BGP configuration mode.

- `update wait-for-convergence` enables the BGP convergence feature.
- `bgp convergence slow-peer time` configures the BGP convergence idle peer timeout value. The default timeout value is 90 seconds.
- `bgp convergence time` configures the BGP convergence timeout value. The default timeout value is 300 seconds.

Different cases for convergence with default timeout configuration

- Convergence Time < 90 seconds after the first peer has joined: This is the best case when all the configured peers have joined and EORs have been received from all peers in less than 90 seconds after the first peer has joined.
- Convergence Time ≥ 90 seconds after the first peer has joined: This is the case when one or more BGP peers have joined within 90 seconds and EORs have been received from all peers within 90 seconds, but there are still some configured peers which have not joined yet. In this case, the convergence is declared after slow-peer timeout is reached.
- Convergence Time > 90 seconds after the first peer has joined: This is the case when one or more BGP peers have joined after 90 seconds, but EORs have not been received from all peers. As soon as EORs are received from all peers which have joined during the first 90 seconds, the convergence is declared.
- Convergence Time ≥ 300 seconds after the first peer has joined: This is the case when EOR is not received till 300 seconds from some of the peers that have joined during 90 seconds after the first peer has joined.

33.2.7.1 Displaying BGP Convergence Status

Use the `show bgp convergence` command to view information about the BGP convergence status, and to know if the convergence timer has started or not.
No peers have joined

- This command displays the output when no peers have joined before convergence.

  switch(config-router-bgp)#show bgp convergence
  BGP Convergence information for VRF: default
  Configured convergence timeout: 00:02:30
  Configured convergence slow peer timeout: 00:00:55
  Convergence based update synchronization is enabled
  Last Bgp convergence event : None
  Bgp convergence state : Not Initiated (Waiting for the first peer to join)
    Convergence timer is not running
    Convergence timeout in use: 00:02:30
    Convergence slow peer timeout in use: 00:00:55
    First peer is not up yet
    All the expected peers are up: no
    All IGP protocols have converged: yes
    Outstanding EORs: 0, Outstanding Keepalives: 0
    Pending Peers: 2
    Total Peers: 2
    Established Peers: 0
    Disabled Peers: 0
    Peers that have not converged yet:
      IPv4 peers:
        201.1.1.1 (Session : Connect)
        202.1.1.1 (Session : Connect)
      IPv6 peers:
        None

First peer has joined

- This command displays the output when the first peer has joined before convergence.

  switch#show bgp convergence
  BGP Convergence information for VRF: default
  Configured convergence timeout: 00:02:30
  Configured convergence slow peer timeout: 00:00:55
  Convergence based update synchronization is enabled
  Last Bgp convergence event 00:00:40 ago
  Bgp convergence state : Pending (Waiting for EORs/Keepalives from peer(s) and IGP convergence)
    Convergence timer running, will expire in 00:01:50
    Convergence timeout in use: 00:02:30
    Convergence slow peer timeout in use: 00:00:55
    First peer came up 00:00:13 ago
    All the expected peers are up: no
    All IGP protocols have converged: yes
    Outstanding EORs: 0, Outstanding Keepalives: 0
    Pending Peers: 1
    Total Peers: 2
    Established Peers: 1
    Disabled Peers: 0
    Peers that have not converged yet:
      IPv4 peers:
        201.1.1.1 (Session : Active)
      IPv6 peers:
        None
Convergence timeout reached

- This command displays the output when the convergence timeout value is reached.

```
switch(config-router-bgp)#show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:02:30
Configured convergence slow peer timeout: 00:00:55
Convergence based update synchronization is enabled
Last Bgp convergence event 00:02:44 ago
Bgp convergence state : Timeout reached
  Time taken to converge 00:02:30
  Pending Peers: 1
  Total Peers: 2
  Established Peers: 1
  Disabled Peers: 0
  Peers that did not converge before local bgp convergence:
    IPv4 peers:
      201.1.1.1 (Session : Active)
      202.1.1.1 (Session : Established)
    IPv6 peers:
      None
```

Converged state

- This command displays the output during the converged state.

```
switch(config-router-bgp)#show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:05:00
Configured convergence slow peer timeout: 00:01:30
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:05 ago
Bgp convergence state : Converged
  Time taken to converge 00:00:02
  First peer came up 00:00:05 ago
  Pending Peers: 0
  Total Peers: 3
  Established Peers: 3
  Disabled Peers: 0
  Peers that did not converge before local bgp convergence:
    IPv4 peers:
      None
    IPv6 peers:
      None
```

33.2.8 Configuring BGP Graceful Shutdown Community

33.2.8.1 Creating a Route-Map Entry that Sets the Community for Graceful Shutdown

The `set community (route-map)` command specifies community attribute modifications to BGP routes.

**Example**

```
switch(config)#route-map map1
switch(config-route-map-map1)#set community GSHUT
switch(config)#exit
```
33.2.8.2 Creating a Route-Map Entry with Matching Preferences on Graceful Shutdown Community

The `ip community-list` command creates and configures a BGP access list that is based on BGP communities.

The `match (route-map)` command creates a route map clause entry that specifies one route filtering condition.

Example

```
switch(config)#ip community-list gshut_list permit GSHUT
switch(config)#route-map map1
switch(config-route-map-map1)#match community gshut_list
switch(config-route-map-map1)#exit
switch(config)#
```

33.2.8.3 Validating the Route-Map

The `show route-map` command displays the contents of the specified route maps.

Example

```
switch#show route-map map1
route-map map1 permit 10
Description:
Match clauses:
Set clauses:
  set community GSHUT
switch#
```

33.2.9 BGP Confederations

BGP confederations allow you to break an autonomous system (AS) into multiple sub-ASs, and then to group the sub-ASs as a confederation.

The sub-ASs exchange iBGP routing information (next-hop, local-preference and MED), but communicate via eBGP.

Configure a BGP confederation by completing the following tasks on each BGP device in the confederation.

- Configuring the local AS number: The local AS number is the membership number in a sub-AS. BGP devices with the same local AS number are identified as members of the same sub-AS. BGP devices always use the local AS number when communicating with other BGP4 devices in the confederation.
- Configuring the confederation ID: The confederation ID is the AS number for those BGP devices that are outside of the confederation. A BGP device outside the confederation is not aware that BGP devices are in multiple sub-ASs. The confederation ID must differ from the sub-AS numbers.
- Configuring the list of sub-AS numbers that are confederation members. Devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP.

Example

- The `router bgp` command enables BGP and configures the router in sub-autonomous system 65050. The `bgp confederation identifier` command specifies confederation 65050 belongs to autonomous system 100.
The neighbors from other autonomous systems within the confederation are treated as special eBGP peers when using the `bgp confederation peers` command.

```
switch(config)#router bgp 65050
switch(config-router-bgp)#bgp confederation identifier 100
switch(config-router-bgp)#bgp confederation peers 65060
```

- The Arista EOS will group the maximum ranges together. In this example, peers 65032 and 65036 are not included in BGP confederation 100.

```
switch(config)#router bgp 65050
switch(config-router-bgp)#bgp confederation identifier 100
switch(config-router-bgp)#bgp confederation peers 65060
```

### 33.2.10 BGP Operational Commands

#### 33.2.10.1 Shutdown

- The `shutdown (BGP)` command disables BGP operations without disrupting the BGP configuration. The `no router bgp` command disables BGP and removes the BGP configuration.

The `no shutdown` command resumes BGP activity.

**Example**

- This command disables BGP activity on the switch.

```
switch(config-router-bgp)#shutdown
switch(config-router-bgp)#
```
• This command resumes BGP activity on the switch.
  
  ```
  switch (config-router-bgp)# no shutdown
  switch (config-router-bgp)#
  ```

33.2.10.2 Clearing the Routing Table and Resetting BGP Sessions

When entered without parameters, the `clear ip bgp` command clears all BGP learned routes from the routing table, reads routes from designated peers, and sends routes required by those peers. Routes that are read or sent are processed through any modified route map or AS-path access list.

Followed by an asterisk (*), it clears the BGP sessions with all BGP peers. To reset the session with a specific peer, enter the peer’s IP address at the end of the command.

**Example**

• This command removes all BGP learned routes from the routing table.

  ```
  switch# clear ip bgp
  switch#
  ```
33.3 **BGP Examples**

This section describes the commands required to configure an iBGP and an eBGP topology.

33.3.1 **Example 1**

Example 1 features an internal BGP link that connects peers in AS 100.

33.3.1.1 **Diagram**

*Figure 33-2* displays BGP Example 1. The BGP link establishes iBGP neighbors in AS 100. Each switch advertises two subnets. In UPDATE packets sent by Switch A, the LOCAL_PREF field is 150. In UPDATE packets sent by Switch B, the LOCAL_PREF field is 75.

*Figure 33-2: BGP Example 1*

33.3.1.2 **Code**

This code configures the Example 1 BGP instance on both switches.

**Step 1** Configure the neighbor addresses.

a Specify the neighbor to Switch A.

```
switchA(config)#router bgp 100
switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 100
```

b Specify the neighbor to Switch B.

```
switchB(config)#router bgp 100
switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
```

**Step 2** Configure the routes to be advertised.

a Advertise Switch A’s routes.

```
switchA(config-router-bgp)#network 10.10.1.0/24
switchA(config-router-bgp)#network 10.10.2.0/24
```

b Advertise Switch B’s routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```
**Step 3** Configure the LOCAL_PREF.

```plaintext
switchA(config-router-bgp)#neighbor 10.100.100.2 export-localpref 150
switchB(config-router-bgp)#neighbor 10.100.100.1 export-localpref 75
```

**Step 4** Modify the hold time and keepalive interval.

```plaintext
switchA(config-router-bgp)#timer bgp 30 90
switchB(config-router-bgp)#timer bgp 30 90
```

### 33.3.2 Example 2

Example 2 creates an external BGP link that connects routers in AS 100 and AS 200.

#### 33.3.2.1 Diagram

*Figure 33-3* displays BGP Example 2. The BGP link connects a switch in AS 100 to a switch in AS 200. Each switch advertises two subnets.

Switch A assigns a local preference of 150 to networks advertised by Switch B. Switch B assigns a local preference of 75 to networks advertised by Switch A.

*Figure 33-3: BGP Example 2*

![BGP Example 2 Diagram](image)

#### 33.3.2.2 Code

This code configures the Example 2 BGP instance on both switches.

**Step 1** Configure the neighbor addresses.

- **a** Specify the neighbor to Switch A.
  ```plaintext
  switchA(config)#router bgp 100
  switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 200
  ```

- **b** Specify the neighbor to Switch B.
  ```plaintext
  switchB(config)#router bgp 200
  switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
  ```
Step 2  Configure the routes to be advertised.
   a  Advertise Switch A's routes.
      switchA(config-router-bgp)#network 10.10.1.0/24
      switchA(config-router-bgp)#network 10.10.2.0/24
   b  Advertise Switch B's routes.
      switchB(config-router-bgp)#network 10.10.3.0/24
      switchB(config-router-bgp)#network 10.10.4.0/24

Step 3  Assign local preference values to routes received from their respective peers.
   switchA(config-router-bgp)#neighbor 10.100.100.2 import-localpref 150
   switchB(config-router-bgp)#neighbor 10.100.100.2 import-localpref 75

Step 4  Modify the hold timer and keepalive interval.
   switchA(config-router-bgp)#timer bgp 30 90
   switchB(config-router-bgp)#timer bgp 30 90
33.4 BGP Commands

Global Configuration Commands

- router bgp
- ip as-path access-list
- ip as-path regex-mode
- ip community-list regexp
- ip community-list
- ip extcommunity-list regexp
- ip extcommunity-list
- ip large-community-list regexp

Router-BGP Configuration Mode (Includes Address-Family Mode)

- address-family
- aggregate-address
- bgp advertise-inactive
- bgp always-compare-med
- bgp bestpath as-path ignore
- bgp bestpath as-path multipath-relax
- bgp bestpath ecmp-fast
- bgp bestpath med confed
- bgp bestpath med missing-as-worst
- bgp bestpath tie-break cluster-list-length
- bgp bestpath tie-break router-id
- bgp client-to-client reflection
- bgp cluster-id
- bgp confederation identifier
- bgp confederation peers
- bgp convergence slow-peer time
- bgp convergence time
- bgp default
- bgp enforce-first-as
- bgp listen range
- bgp log-neighbor-changes
- bgp redistribute-internal (BGP)
- distance bgp
- dynamic peer max
- graceful-restart stalepath-time
- graceful-restart-helper
- maximum paths (BGP)
- match as-range
- no neighbor
- neighbor activate
- neighbor allowas-in
- neighbor auto-local-addr
- neighbor default-originate
- neighbor description
- neighbor ebgp-multihop
- neighbor enforce-first-as
- neighbor export-localpref
Chapter 33: Border Gateway Protocol (BGP)  

**BGP Commands**

- neighbor graceful-restart-helper
- neighbor import-localpref
- neighbor local-as
- neighbor local-v4-addr
- neighbor local-v6-addr
- neighbor maximum-routes
- neighbor next-hop-peer
- neighbor next-hop-self
- neighbor out-delay
- neighbor passive
- neighbor password
- neighbor peer group (create)
- neighbor peer-group (neighbor assignment)
- neighbor remote-as
- neighbor remove-private-as
- neighbor rib-in pre-policy retain
- neighbor route-map (BGP)
- neighbor route-reflector-client
- neighbor send-community
- neighbor shutdown
- neighbor timers
- neighbor ttl maximum-hops
- neighbor update-source
- neighbor weight
- network (BGP)
- peer-filter
- rd (Router-BGP VRF and VNI Configuration Modes)
- redistribute (BGP)
- router-id (BGP)
- shutdown (BGP)
- timers bgp
- update wait-for-convergence
- vrf

**Route Map Configuration Mode**

- set as-path match

**Clear Commands – Privileged EXEC Mode**

- clear ip bgp
- clear ip bgp neighbor *
- clear ipv6 bgp
- clear ipv6 bgp neighbor *

**Display Commands – EXEC Mode**

- show bgp convergence
- show bgp instance
- show ip as-path access-list
- show ip bgp
- show ip bgp community
- show ip bgp neighbors
- show ip bgp neighbors (route type)
- show ip bgp neighbors (route-type) community
- show ip bgp neighbors regexp
- show ip bgp paths
- show ip bgp peer-group
- show ip bgp regexp
- show ip bgp summary
- show ip community-list
- show ip extcommunity-list
- show ipv6 bgp
- show ipv6 bgp match community
- show ipv6 bgp peers
- show ipv6 bgp peers (route type)
- show ipv6 bgp peers (route type) community
- show ipv6 bgp peers regexp
- show ipv6 bgp regexp
- show ipv6 bgp summary
- show peer-filter
address-family

The address-family command places the switch in address-family configuration mode to configure the address family setting of addresses configured as BGP neighbors. Address-family configuration mode is not a group change mode; running-config is changed immediately after commands are executed. The exit command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

Running-config displays address family commands in sub-blocks of the BGP configuration. The following commands are available in address family configuration mode:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.
- **neighbor default-originate** advertises a default route to the specified BGP neighbor.
- **neighbor route-map (BGP)** applies a route map to the specified BGP route.
- **network (BGP)** specifies a network for advertisement through UPDATE packets to BGP peers.

The no address-family and default address-family commands delete the specified address-family from running-config by removing all commands previously configured in the corresponding address-family mode.

The exit command returns the switch to router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp ADDRESS_TYPE
no bgp ADDRESS_TYPE
default bgp ADDRESS_TYPE
```

Parameters

- **ADDRESS_FAMILY** Address family affected by subsequent commands. Options include:
  
  - ipv4 IPv4 unicast
  - ipv6 IPv6 unicast

Example

- These commands enter address family mode for IPv6-unicast, insert a command, then exit the mode:

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#address-family ipv6
  switch(config-router-bgp-af)#neighbor 172.10.1.1 activate
  switch(config-router-bgp-af)#exit
  switch(config-router-bgp-af)#
  ```
aggregate-address

The `aggregate-address` command creates an aggregate route in the Border Gateway Protocol (BGP) database. Aggregate routes combine the characteristics of multiple routes into a single route that the switch advertises. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregate routes are advertised only after they are redistributed.

The advertised address of the aggregate is entered as an IP subnet; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.

**Important!** Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

Command options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Command options affect the following aggregate routing attributes:

- **AS_PATH attribute inclusion:** the `as-set` option controls the aggregate route’s AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

  When the command includes `as-set`, the aggregate route’s AS_SET attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the AS_SET.

  When the command does not include `as-set`, the aggregate route’s ATOMIC_AGGREGATE attribute is set and the AS_PATH attribute does not include AS numbers of contributing routes.

- **Attribute assignment:** The `attribute-map` option assigns attributes contained in set commands in a specified route map’s lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route’s attributes by the switch.

- **Route suppression:** The `summary-only` option suppresses the advertisement of the contributor routes that comprise the aggregate.

- **Contributor filtering:** The `match-map` option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

The `no aggregate-address` and `default aggregate-address` commands remove the corresponding aggregate-address command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
aggregate-address AGGREGATE_NET [AS_SET] [SUMMARY] [ATTRIBUTE_MAP] [MATCH_MAP]
no aggregate-address AGGREGATE_NET
default aggregate-address AGGREGATE_NET
```

**Parameters**

- `AGGREGATE_NET` aggregate route IP address. Options include:
  - `netv4_addr` IPv4 subnet address (CIDR or address-mask notation).
  - `netv6_addr` IPv6 subnet address (CIDR notation).
• **AS_SET** controls AS_PATH attribute values associated with aggregate route. Options include:
  - <no parameter> ATOMIC_AGGREGATE attribute is set. Route contains no AS_PATH data.
  - as-set route includes AS_PATH information from contributor routes as AS_SET attributes.

• **SUMMARY** controls advertisement of contributor routes. Options include:
  - <no parameter> contributor and aggregate routes are advertised.
  - summary-only contributor routes are not advertised.

• **ATTRIBUTE_MAP** controls attribute assignments to the aggregate route. Options include:
  - <no parameter> attribute values are not assigned to route.
  - attribute-map map_name assigns attribute values in set commands of the map’s permit clauses. Deny clauses and match commands in permit clauses are ignored.

• **MATCH_MAP** filters contributors to the aggregate route. Options include:
  - <no parameter> no contributors are filtered.
  - match-map map_name filters contributor routes using the named match-map.

**Examples**

• These commands create an aggregate route (10.16.48.0/20) from the contributor routes 10.16.48.0/23, 10.16.50.0/23, 10.16.52.0/23, and 10.16.54.0/23. The aggregate route includes the AS_PATH information from the contributor routes.

  switch(config)#router bgp 1
  switch(config-router-bgp)#aggregate-address 10.16.48.0/20 as-set
  switch(config-router-bgp)#exit
  switch(config)#

• These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

  switch(config)#route-map map1 permit 10
  switch(config-route-map-map1)#set community 45
  switch(config-route-map-map1)#exit
  switch(config)#router bgp 1
  switch(config-router-bgp)#aggregate-address 10.16.48.0/20 attribute-map map1
  switch(config-router-bgp)#exit
  switch(config)#

• These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

  switch(config)#route-map matchmap permit 10
  switch(config-route-map-matchmap)#match ip address prefix-list agglist
  switch(config-route-map-matchmap)#exit
  switch(config)#router bgp 1
  switch(config-router-bgp)#aggregate-address 1.1.0.0/16 match-map matchmap
**bgp advertise-inactive**

By default, BGP will advertise only those routes that are active in the switch’s RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The `bgp advertise-inactive` command configures BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

The `no bgp advertise-inactive` and `default bgp advertise-inactive` commands restore the default BGP behavior (advertising only active routes) by removing the corresponding `bgp advertise-inactive` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```plaintext
bgp advertise-inactive
no bgp advertise-inactive
default bgp advertise-inactive
```

**Example**

- These commands configure BGP to advertise inactive routes.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp advertise-inactive
switch(config-router-bgp)#
```
bgp always-compare-med

The `bgp always-compare-med` command configures the switch to always consider multi-exit discriminator (MED) values (also known as “metric”) in best-path selection. By default, this function is disabled, and MED values are compared only if two paths have the same neighbor AS.

When there are two or more links between autonomous systems, MED values may be set by a router in the originating AS to give preferences to certain routes. In comparing MED values, the lower value is preferred.

The `no bgp always-compare-med` and `default bgp always-compare-med` commands restore the default behavior of comparing MED values only on paths with the same neighbor AS.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

- `bgp always-compare-med`
- `no bgp always-compare-med`
- `default bgp always-compare-med`

**Related Commands**

- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

**Example**

These commands configure BGP to always consider MED values in best-path comparisons.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp always-compare-med
switch(config-router-bgp)#
```
bgp bestpath as-path ignore

The `bgp bestpath as-path ignore` command configures BGP to ignore the length of the autonomous system (AS) path when comparing routes. This behavior is disabled by default. Normally, the switch compares AS paths as the third step in the best-path selection process (see Best-Path Selection), preferring the route with the shorter AS path.

The `no bgp bestpath as-path ignore` and `default bgp bestpath as-path ignore` commands restore the default behavior of considering AS path length in route comparisons.

Command Mode
- Router-BGP Configuration

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

Related Commands
- `bgp always-compare-med`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

Example
- These commands configure BGP to ignore AS path lengths when comparing routes.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath as-path ignore
switch(config-router-bgp)#
```
bgp bestpath as-path multipath-relax

The `bgp bestpath as-path multipath-relax` command allows multiple eBGP routes to a destination to be considered equal in ECMP if their AS paths are the same length despite having different autonomous systems in those paths. The `no bgp bestpath as-path multipath-relax` command configures best-path selection to consider two paths `unequal` if their AS path contents are different, and prefers the first path received.

Multipath-relax is enabled by default. The `bgp bestpath as-path multipath-relax` and `default bgp bestpath as-path multipath-relax` commands restore the default behavior by removing the corresponding `no bgp bestpath as-path multipath-relax` command from `running-config`.

For BGP to support equal cost multipath (ECMP) routing, the `maximum paths (BGP)` command must be issued in router-BGP configuration mode.

**Command Mode**
- Router-BGP Configuration

**Command Syntax**

```
bgp bestpath as-path multipath-relax
no bgp bestpath as-path multipath-relax
default bgp bestpath as-path multipath-relax
```

**Related Commands**
- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

**Example**

```
These commands configure BGP best-path selection to consider routes unequal if the contents of their AS paths differ.

switch(config)#router bgp 64500
switch(config-router-bgp)#no bgp bestpath as-path multipath-relax
switch(config-router-bgp)#
```
**bgp bestpath ecmp-fast**

By default, within an ECMP group the BGP best-path selection process prefers the active path (the first path received by the switch) unless a relevant tie-breaker is enabled. The `no bgp bestpath ecmp-fast` command causes the best-path selection process to ignore order of arrival and continue evaluating paths on other criteria.

The `bgp bestpath ecmp-fast` and `default bgp bestpath ecmp-fast` commands restore the default behavior by removing the corresponding `no bgp bestpath ecmp-fast` command from `running-config`.

**Command Mode**

- Router-BGP Configuration

**Command Syntax**

```
bgp bestpath ecmp-fast
no bgp bestpath ecmp-fast
default bgp bestpath ecmp-fast
```

**Related Commands**

- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

**Example**

- These commands configure BGP to ignore order of arrival in best-path comparisons of paths within an ECMP group.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#no bgp bestpath ecmp-fast
switch(config-router-bgp)#
```
bgp bestpath med confed

By default, paths originating within the same confederation as the switch and received from confederation peers do not have their multi-exit discriminator (MED) values compared as part of the best-path selection process. The `bgp bestpath med confed` command causes comparison of MED values in such routes. To ensure that MED values are considered in the best-path selection process for all routes received, use the `bgp always-compare-med` command.

The `no bgp bestpath med confed` and `default bgp bestpath med confed` commands restore the default behavior by removing the corresponding `bgp bestpath ecmp-fast` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp bestpath med confed [missing-as-worst]
nobgp bestpath med confed [missing-as-worst]
default bgp bestpath med confed [missing-as-worst]
```

**Related Commands**

- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

**Parameters**

- `missing-as-worst` By default, best-path selection considers a missing MED value to be 0, so paths with missing MED values will be preferred. This option reverses the behavior in comparisons of routes originating within the same confederation as the switch, treating a missing MED as having the highest (least preferred) value.

**Note**

The `bgp bestpath med missing-as-worst` command controls how best-path selection treats missing MED values for all routes received, and, if configured, overrides the `missing-as-worst` option of this command.

**Example**

- These commands configure the BGP best-path selection process to consider MED values in comparisons between routes originating within the same confederation as the switch.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath med confed
switch(config-router-bgp)#
```
**bgp bestpath med missing-as-worst**

By default, BGP best-path selection considers a missing MED value to be 0, so paths with missing MED values will be preferred. The `bgp bestpath med missing-as-worst` command reverses the behavior, treating a missing MED as having the highest (least preferred) value.

The `no bgp bestpath med missing-as-worst` and `default bgp bestpath med missing-as-worst` commands restore the default behavior (giving preference to missing MED values) by removing the corresponding `bgp bestpath med missing-as-worst` command from `running-config`.

---

**Note**

This command overrides the `missing-as-worst` setting of the `bgp bestpath med confed` command.

---

**Command Mode**

Router-BGP Configuration

**Command Syntax**

- `bgp bestpath med missing-as-worst`
- `no bgp bestpath med missing-as-worst`
- `default bgp bestpath med missing-as-worst`

**Related Commands**

- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath tie-break cluster-list-length`
- `bgp bestpath tie-break router-id`

**Example**

- These commands configure the BGP best-path selection process to consider a missing MED value to be considered highest (least preferred) in MED comparisons for all routes received.

```plaintext
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath med missing-as-worst
switch(config-router-bgp)#
```
**bgp bestpath tie-break cluster-list-length**

The `bgp bestpath tie-break cluster-list-length` command causes the best-path selection process to prefer the multipath route with the shortest CLUSTER_LIST length in case of a tie in step 10. The cluster list length is assumed to be 0 if the route doesn’t carry a CLUSTER_LIST attribute.

The `no bgp bestpath tie-break cluster-list-length` and `default bgp bestpath tie-break cluster-list-length` commands restore the default behavior by removing the associated `bgp bestpath tie-break cluster-list-length` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp bestpath tie-break cluster-list-length
no bgp bestpath tie-break cluster-list-length
default bgp bestpath tie-break cluster-list-length
```

**Related Commands**

- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break router-id`

**Example**

- These commands configure the BGP selection process to prefer the multipath route with the shortest CLUSTER_LIST length in case of a tie.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath tie-break cluster-list-length
switch(config-router-bgp)#
```
**bgp bestpath tie-break router-id**

The `bgp bestpath tie-break router-id` command causes the best-path selection process to prefer the multipath route with the lowest ROUTER_ID in case of a tie in step 10. If the route is a reflected route (i.e., if it contains route reflector attributes), the process will use the ORIGINATOR_ID as the ROUTER_ID for comparison. This behavior is disabled by default.

The `no bgp bestpath tie-break router-id` and `default bgp bestpath tie-break router-id` commands restore the default behavior by removing the associated `bgp bestpath tie-break router-id` command from `running-config`.

**Command Mode**
- Router-BGP Configuration

**Command Syntax**
- `bgp bestpath tie-break router-id`
- `no bgp bestpath tie-break router-id`
- `default bgp bestpath tie-break router-id`

**Related Commands**
- `bgp always-compare-med`
- `bgp bestpath as-path ignore`
- `bgp bestpath as-path multipath-relax`
- `bgp bestpath ecmp-fast`
- `bgp bestpath med confed`
- `bgp bestpath med missing-as-worst`
- `bgp bestpath tie-break cluster-list-length`

**Example**
- These commands configure the best-path selection process to prefer the multipath route with the lowest ROUTER_ID in case of a tie.
  ```
  switch(config)#router bgp 64500
  switch(config-router-bgp)#bgp bestpath tie-break router-id
  switch(config-router-bgp)#
  ```
bgp client-to-client reflection

By default, routes received from a route reflector client and selected as best routes are propagated to all BGP peers, including other route reflector clients. If the clients are fully meshed, however, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled.

The no bgp client-to-client reflection command disables client-to-client reflection.

The bgp client-to-client reflection and default bgp client-to-client reflection commands restore the default behavior by removing the no bgp client-to-client reflection command from running-config.

Command Mode

Router-BGP Configuration

Command Syntax

  bgp client-to-client reflection
  no bgp client-to-client reflection
  default bgp client-to-client reflection

Example

  • This command disables client-to-client reflection on the switch.

    switch(config)#router bgp 1
    switch(config-router-bgp)#no bgp client-to-client reflection
    switch(config-router-bgp)#
**bgp cluster-id**

When using route reflectors, an AS is divided into clusters. A cluster consists of one or more route reflectors and a group of clients to which they re-advertise route information, and for redundancy a single cluster may contain multiple route reflectors. Each route reflector has a cluster ID. If the cluster has only one route reflector the cluster ID is its router ID, but if a cluster has multiple route reflectors a 4-byte cluster ID must be assigned to all route reflectors in the cluster. All must be configured with the same cluster ID to allow them to identify updates from the cluster’s other route reflectors.

The `bgp cluster-id` command configures the cluster ID in a cluster with multiple route reflectors.

The `no bgp cluster-id` and `default bgp cluster-id` commands remove the cluster ID by removing the corresponding `bgp cluster-id` command from `running-config`. Do not remove the cluster ID if there are multiple route reflectors in the cluster.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp cluster-id ID_NUM
no bgp cluster-id
default bgp cluster-id
```

**Parameters**

- `ID_NUM` cluster ID shared by all route reflectors in the cluster (32-bit dotted-decimal notation).
  
  Options include:
  
  - `0.0.0.1` to `255.255.255.255` valid cluster ID number.
  - `0.0.0.0` removes the cluster-ID from the switch. Equivalent to `no bgp cluster-id` command.

**Example**

- This command sets the cluster ID for the switch to 172.22.30.101.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```
### bgp confederation identifier

The `bgp confederation identifier` command configures the confederation identifier. Confederation can reduce the number of iBGP connections in a large AS domain. The AS domain is divided into several smaller sub-ASs, and each sub-AS remains fully connected. Devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP.

The `no bgp confederation identifier` and `default bgp confederation identifier` commands remove the `bgp confederation identifier` command from `running-config`.

#### Command Mode
- Router-BGP Configuration

#### Command Syntax

```
bgp confederation identifier as_number
no bgp confederation identifier
default bgp confederation identifier
```

#### Parameters
- `as_number` the ID of BGP AS confederation. Value ranges from 1 to 4294967295.

#### Example
- This command sets the BGP confederation identifier to 9.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp confederation identifier 9
switch(config-router-bgp)#
```
bgp confederation peers

The `bgp confederation peers` command configures a confederation consisting of sub-ASs.

Before this command is executed, the confederation ID should be configured by the `bgp confederation identifier` command. Otherwise this configuration is invalid. The configured ASs in this command are inside the confederation and each AS uses a fully meshed network. The confederation appears as a single AS to the devices outside it.

The `no bgp confederation peers` and `default bgp confederation peers` commands delete the specified sub-AS from the confederation by removing the corresponding `bgp confederation peers` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp confederation peers as_range
no bgp confederation peers as_range
default bgp confederation peers as_range
```

**Parameters**

- `as_range` the Sub-AS number.
  
  `as_range` formats include number (from 1 to 4294967295), number range, or comma-delimited list of numbers and ranges.

**Example**

- This command configures the confederation that contains AS 1000 and 1002.

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#bgp confederation peers 1000 1002
  switch(config-router-bgp)#
  ```
**bgp convergence slow-peer time**

The `bgp convergence slow-peer time` command configures the idle peer time to wait for the slow peers to establish a session in a BGP convergence state.

The `no bgp convergence slow-peer time` command disables the inheritance of the configuration from the global BGP configuration mode. The `default bgp convergence slow-peer time` command sets the timeout value to the default value.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp convergence slow-peer time timeout_range
no bgp convergence slow-peer time
default bgp convergence slow-peer time
```

**Parameters**

- `time timeout_range` The maximum time to wait for the slow peers to establish a session connection. The time ranges from 1 to 3600 seconds. The default value is 90 seconds.

**Example**

- This command configures an idle peer timeout of 40 seconds to wait before establishing a session.
  
  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#bgp convergence slow-peer time 40
  switch(config-router-bgp)#
  ```
**bgp convergence time**

The `bgp convergence time` command configures the time to wait before the BGP convergence starts in a session.

The `no bgp convergence time` command removes the configured convergence time to wait. The `default bgp convergence time` command sets the timeout value to the default value.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp convergence time timeout_range
no bgp convergence time
default bgp convergence time
```

**Parameters**

- **time timeout_range** The maximum time to wait for the BGP convergence. The time ranges from 1 to 3600 seconds. The default value is 300 seconds.

**Example**

- This command configures a convergence time of 200 seconds to wait before establishing a session.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp convergence time 200
switch(config-router-bgp)#
```
bgp default

The `bgp default` command configures the default address family activation level of all addresses configured as BGP neighbors. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The following commands configure default address family activation levels for addresses configured as BGP neighbors:

- `bgp default ipv4-unicast`: all addresses are IPv4 address family active.
- `no bgp default ipv4-unicast`: all addresses are not IPv4 address family active.
- `bgp default ipv6-unicast`: all addresses are IPv6 address family active.
- `no bgp default ipv6-unicast`: all addresses are not IPv6 address family active.
- `bgp default ipv4-unicast transport ipv6`: all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.

**Note**

If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command `bgp default ipv4-unicast transport ipv6`. For an individual neighbor, use the `neighbor activate` command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

The activation state of an individual BGP neighbor address is configured by the `neighbor activate` commands. The `neighbor activate` command overrides the address’s default activation state for the address family configuration mode in which the command is issued:

- `neighbor activate`: the specified address is active.
- `no neighbor activate`: the specified address is not active.

The `default-default address family` activation state defines address family activation level of all addresses configured as BGP neighbors when `running-config` does not contain any `bgp default` commands. The default state of the BGP default activation level varies by address family:

- `IPv4 address family`: all BGP addresses are IPv4 address family active.
- `IPv6 address family`: all BGP addresses are not IPv6 address family active.

The `default bgp default` command restores the default-default activation setting for BGP neighbor addresses in the specified address family:

- `default bgp ipv4-unicast` is equivalent to `bgp ipv4-unicast`
- `default bgp ipv6-unicast` is equivalent to `no bgp ipv6-unicast`

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
bgp default ADDRESS_FAMILY
no bgp default ADDRESS_FAMILY
default bgp default ADDRESS_FAMILY
```

**Parameters**

- `ADDRESS_FAMILY`: BGP address family. Options include:
• **ipv4-unicast** IPv4-unicast peering sessions.
• **ipv6-unicast** IPv6-unicast peering sessions.

**Example**

These commands configure the switch to configure all BGP neighbor addresses as IPv4 address family active and IPv6 address family active.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp default ipv4-unicast
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#show active
```

```
router bgp 65533
  bgp log-neighbor-changes
  distance bgp 20 200 200
  neighbor 172.23.254.2 remote-as 65533
  neighbor 172.41.254.78 remote-as 65534
  neighbor 2001:0DB8:52a4:fe01::2 remote-as 65533
  neighbor 2001:0DB8:52a4:fe4c::1 out-delay 10
```

The show active command does not display the `bgp default ipv4-unicast` command because it is the default setting for IPv4 peering sessions.
bogp enforce-first-as

The **bogp enforce-first-as** command causes a forced comparison of the first autonomous system (AS) in the AS path of eBGP routes received from BGP neighbors to the configured remote external peer autonomous system number (ASN). Updates from eBGP peers that do not include that ASN as the first item in the AS path (in the AS_PATH attribute) are discarded.

This behavior is enabled by default upon BGP configuration, and disabled globally by the **no** form of this command. To configure enforce-first-as for an individual neighbor or peer group, use the **neighbor enforce-first-as** command.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

- `bogp enforce-first-as`
- `default bogp enforce-first-as`
- `no bogp enforce-first-as`

**Example**

- This command configures BGP to enforce the first AS globally.

```
switch(config-router-bgp)#bogp enforce-first-as
switch(config-router-bgp)#
```
**bgp listen range**

The **bgp listen range** command identifies the BGP peering request from a range of IPv4 or IPv6 address, and names the dynamic peer group to which those peers belong to. To create a static peer group, use the neighbor peer group (create) command.

The request can be from a single AS number or from a range of AS numbers configured. To accept the peering request from single ASN use the **remote-as** option, and to accept request from multiple ASNs use the **peer-filter** option.

Members of dynamic peer group are configured in groups and not as individuals. Once a new peer group is created with a group name, the group name is then used as a parameter by the following **neighbor** commands:

- neighbor ebgp-multihop
- neighbor import-localpref
- neighbor maximum-routes
- neighbor route-map (BGP)
- neighbor timers
- neighbor update-source.

The **no bgp listen range** and **default bgp listen range** commands remove the dynamic peer group by deleting the corresponding command from running-config. To remove a static peer group, use the **no neighbor** command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

**Command Mode**
Router-BGP Configuration

**Command Syntax**

```
bgp listen range NET_ADDRESS peer-group group_name [remote-as as_number | peer-filter filter_name]
no bgp listen range NET_ADDRESS peer-group group_name
default bgp listen range NET_ADDRESS peer-group group_name
```

**Parameters**

- **NET_ADDRESS**  IP address range. Entry options include:
  - **IPv4 subnet**  IPv4 subnet (CIDR notation).
  - **IPv4 address mask subnet**  IPv4 subnet (dotted decimal notation).
  - **IPv6_prefix**  IPv6 subnet (dotted decimal notation).
- **group_name**  name of the peer group.
- **as_number**  the autonomous system number, ranges from 1 to 4294967295.
- **filter_name**  name of the peer filter.

**Example**

- These commands create a dynamic peer group called “brazil” in AS 5 which accepts peering requests from the 192.168.6.0/24 subnet.
  
  switch(config)#router bgp 1
  switch(config-router-bgp)#bgp listen range 192.168.6.0/24 peer-group brazil
  remote-as 5
These commands create a dynamic peer group called “brazil” in a range of ASNs, which accepts peering requests from the 192.0.2.0/24 subnet. The range of AS numbers is defined by peer filter option.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 192.0.2.0/24 peer-group brazil peer-filter group-1
```
**bgp log-neighbor-changes**

The **bgp log-neighbor-changes** command configures the switch to generate a log message when a BGP peer enters or exits the Established state. This is the default behavior.

The **no bgp log-neighbor-changes** command disables the generation of these log messages. The **default bgp log-neighbor-changes** command enables the generation of these log messages.

**Command Mode**
- Router-BGP Configuration

**Command Syntax**

```
bgp log-neighbor-changes
no bgp log-neighbor-changes
default bgp log-neighbor-changes
```

**Example**

- These commands configure the switch to generate a message when a BGP peer enters or exits the *established* state.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp log-neighbor-changes
switch(config-router-bgp)#
```
bgp redistribute-internal (BGP)

The `bgp redistribute-internal` command enables the redistribution of iBGP routes into an interior gateway protocol (IGP).

The `no bgp redistribute-internal` and `default bgp redistribute-internal` commands disable route redistribution from the specified domain by removing the corresponding `bgp redistribute-internal` command from `running-config`.

Command Mode
- Router-BGP Configuration
- Router-BGP Address-Family Configuration

Command Syntax
- `bgp redistribute internal`
- `no bgp redistribute internal`
- `default bgp redistribute internal`

Example
- This command redistributes internal BGP routes.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#bgp redistribute-internal
  switch(config-router-bgp)#
  ```
clear ip bgp

The clear ip bgp command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required. This command can also clear the switch’s BGP sessions with its peers.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch’s BGP sessions with its peers.

Command Mode
Privileged EXEC

Command Syntax
```
clear ip bgp [PEERS] [RESET_TYPE] [DATA_FLOW] [VRF_INSTANCE]
```

Parameters
- **PEERS** specifies targeted BGP peers. Options include:
  - <no parameters> all IPv4 and IPv6 peers.
  - * all IPv4 and IPv6 peers.
  - ipv4_addr the IPv4 peer with the specified IPv4 address.
  - ipv6_addr the IPv6 peer with the specified IPv6 address.
  - intrf_ipv6_addr the peer using the specified IPv6 link-local address.
  - peer-group peer_grp_name the peers using the specified BGP peer group.
- **RESET_TYPE** specifies the method used to reset routes. Options include:
  - <no parameters> performs a hard reset that terminates current BGP sessions and recreates the local routing information base.
  - soft performs a soft reset that maintains current BGP sessions and reconfigures the local routing information base using stored routes.
- **DATA_FLOW** restricts soft reset to inbound or outbound routes. Hard reset is bidirectional. Options include:
  - <no parameters> resets inbound and outbound routes.
  - in resets inbound peer routes.
  - out resets outbound peer routes.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - <no parameters> resets matching peers in the context-active VRF.
  - vrf_name resets matching peers in the specified VRF.
  - all resets matching peers in all VRFs.
  - default resets matching peers in the default VRF.

Related Commands
- clear ip bgp counters
- clear ip bgp errors
- clear ip bgp neighbor *

Guidelines
Use the clear ip bgp command after changing any of the following BGP attributes:
- Weights
• Distribution lists
• Timers
• Administrative distance

Examples
• This command performs a hard reset of all IPv4 and IPv6 peers in the context-active VRF.
  switch# clear ip bgp
  switch#

• This command has the same behavior as in clear ip bgp command.
  switch# clear ip bgp *
  switch#
clear ip bgp counters

The clear ip bgp counters command resets general statistics of peers. It primarily consists of message related counts.

Command Mode
Privileged EXEC

Command Syntax
  clear ip bgp [PEERS] counters [VRF_INSTANCES]

Parameters
- **PEERS** specifies targeted BGP peers. Options include:
  - <no parameters> all IPv4 and IPv6 peers.
  - * all IPv4 and IPv6 peers.
  - ipv4_addr the IPv4 peer with the specified IPv4 address.
  - ipv6_addr the IPv6 peer with the specified IPv6 address.
  - intrf_ipv6_addr the peer using the specified IPv6 link-local address.
  - peer-group peer_grp_name the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - <no parameters> resets matching peers in the context-active VRF.
  - vrf_name resets matching peers in the specified VRF.
  - all resets matching peers in all VRFs.
  - default resets matching peers in the default VRF.

Related Commands
- clear ip bgp
- clear ip bgp errors
- clear ipv6 bgp counters

Example
- This command resets general statistics of all IPv4 and IPv6 peers in the context-active VRF.
  switch#clear ip bgp counters
  ! Clearing all IPv4 and IPv6 peering sessions
  switch#
clear ip bgp errors

The `clear ip bgp errors` command resets the error statistics and history of peers. Peer general statistics primarily consists of notification errors, socket errors, and update errors.

**Command Mode**
Privileged EXEC

**Command Syntax**
```
clear ip bgp [PEERS] errors [VRF_INSTANCES]
```

**Parameters**
- **PEERS** specifies targeted BGP peers. Options include:
  - `<no parameters>` all IPv4 and IPv6 peers.
  - `*` all IPv4 and IPv6 peers.
  - `ipv4_addr` the IPv4 peer with the specified IPv4 address.
  - `ipv6_addr` the IPv6 peer with the specified IPv6 address.
  - `intrf_ipv6_addr` the peer using the specified IPv6 link-local address.
  - `peer-group peer_grp_name` the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - `<no parameters>` resets matching peers in the context-active VRF.
  - `vrf_name` resets matching peers in the specified VRF.
  - `all` resets matching peers in all VRFs.
  - `default` resets matching peers in the default VRF.

**Related Commands**
- `clear ip bgp`
- `clear ip bgp counters`
- `clear ipv6 bgp errors`

**Example**
- This command resets the error statistics of all IPv4 and IPv6 peers in the context-active VRF.
```
switch#clear ip bgp errors
! Clearing all IPv4 and IPv6 peering sessions
switch#
```
**clear ip bgp neighbor ***

The `clear ip bgp neighbor *` command clears BGP neighbors belonging to the IPv4 transport address family. To clear BGP neighbors in the IPv6 transport address family, use the `clear ipv6 bgp neighbor *` command.

**Command Mode**
Privileged EXEC

**Command Syntax**
```
clear ip bgp neighbor * [VRF_INSTANCE]
```

**Parameters**
- `VRF_INSTANCE` specifies VRF instance for which IPv4 transport address family BGP neighbors will be cleared. Options include:
  - `<no parameter>` clears IPv4 BGP neighbors in the context-active VRF.
  - `vrf vrf_name` clears IPv4 BGP neighbors in the specified VRF.
  - `vrf all` clears IPv4 BGP neighbors in all VRFs.
  - `vrf default` clears IPv4 BGP neighbors in the default VRF.

**Related Commands**
- `clear ip bgp`
- `clear ip bgp counters`
- `clear ip bgp errors`

**Examples**
- This command clears all IPv4 BGP neighbors in the context-active VRF.
  
  ```
  switch# clear ip bgp neighbor *
  switch#
  ```

- This command clears all IPv4 BGP neighbors in VRF “purple.”
  
  ```
  switch# clear ip bgp neighbor * vrf purple
  switch#
  ```
clear ipv6 bgp

The clear ipv6 bgp command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required. This command can also clear the switch’s BGP sessions with its peers.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch’s BGP sessions with its peers.

Command Mode
Privileged EXEC

Command Syntax

```
clear ipv6 bgp [PEERS] [RESET_TYPE] [DATA_FLOW] [VRF_INSTANCE]
```

Parameters

- **PEERS** specifies targeted BGP peers. Options include:
  - <no parameters> all IPv4 and IPv6 peers.
  - * all IPv4 and IPv6 peers.
  - ipv4_addr the IPv4 peer with the specified IPv4 address.
  - ipv6_addr the IPv6 peer with the specified IPv6 address.
  - intrf_ipv6_addr the peer using the specified IPv6 link-local address.
  - peer-group peer_grp_name the peers using the specified BGP peer group.

- **RESET_TYPE** specifies the method used to reset routes. Options include:
  - <no parameters> performs a hard reset that terminates current BGP sessions and recreates the local routing information base.
  - soft performs a soft reset that maintains current BGP sessions and reconfigures the local routing information base using stored routes.

- **DATA_FLOW** restricts soft reset to inbound or outbound routes. Hard reset is bidirectional. Options include:
  - <no parameters> resets inbound and outbound routes.
  - in resets inbound peer routes.
  - out resets outbound peer routes.

- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - <no parameters> resets matching peers in the context-active VRF.
  - vrf_name resets matching peers in the specified VRF.
  - all resets matching peers in all VRFs.
  - default resets matching peers in the default VRF.

Related Commands

- clear ipv6 bgp counters
- clear ipv6 bgp errors
- clear ipv6 bgp neighbor *

Guidelines

Use the clear ipv6 bgp command after changing any of the following BGP attributes:

- Weights
- Distribution lists
- Timers
- Administrative distance

**Examples**

- This command performs a hard reset of all IPv4 and IPv6 peers in the context-active VRF.
  
  ```
  switch# clear ipv6 bgp
  switch#
  ```

- This command has the same behavior as in `clear ipv6 bgp` command.
  
  ```
  switch# clear ipv6 bgp *
  switch#
  ```
clear ipv6 bgp counters

The **clear ipv6 bgp counters** command resets general statistics of peers. Peer general statistics primarily consists of message related counts.

**Command Mode**

Privileged EXEC

**Command Syntax**

```
clear ipv6 bgp [PEERS] counters [VRF_INSTANCES]
```

**Parameters**

- **PEERS** specifies targeted BGP peers. Options include:
  - <no parameters> all IPv4 and IPv6 peers.
  - * all IPv4 and IPv6 peers.
  - ipv4_addr the IPv4 peer with the specified IPv4 address.
  - ipv6_addr the IPv6 peer with the specified IPv6 address.
  - intrf_ipv6_addr the peer using the specified IPv6 link-local address.
  - peer-group peer_grp_name the peers using the specified BGP peer group.

- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - <no parameters> resets matching peers in the context-active VRF.
  - vrf_name resets matching peers in the specified VRF.
  - all resets matching peers in all VRFs.
  - default resets matching peers in the default VRF.

**Related Commands**

- clear ip bgp counters
- clear ipv6 bgp
- clear ipv6 bgp errors

**Example**

- This command resets general statistics of all IPv4 and IPv6 peers in the context-active VRF.
  ```
  switch#clear ipv6 bgp counters
  ! Clearing all IPv4 and IPv6 peering sessions
  switch#
  ```
clear ipv6 bgp errors

The `clear ipv6 bgp errors` command resets the error statistics and history of peers. Peer error statistics primarily consists of notification errors, socket errors, and update errors.

**Command Mode**
Privileged EXEC

**Command Syntax**
```
clear ipv6 bgp [PEERS] errors [VRF_INSTANCES]
```

**Parameters**
- **PEERS** specifies targeted BGP peers. Options include:
  - `<no parameters>` all IPv4 and IPv6 peers.
  - `*` all IPv4 and IPv6 peers.
  - `ipv4_addr` the IPv4 peer with the specified IPv4 address.
  - `ipv6_addr` the IPv6 peer with the specified IPv6 address.
  - `intrf_ipv6_addr` the peer using the specified IPv6 link-local address.
  - `peer-group peer_grp_name` the peers using the specified BGP peer group.
- **VRF_INSTANCES** specifies the VRF(s) examined for BGP peers. Options include:
  - `<no parameters>` resets matching peers in the context-active VRF.
  - `vrf_name` resets matching peers in the specified VRF.
  - `all` resets matching peers in all VRFs.
  - `default` resets matching peers in the default VRF.

**Related Commands**
- `clear ip bgp errors`
- `clear ipv6 bgp`
- `clear ipv6 bgp counters`

**Example**
- This command resets the error statistics of all IPv4 and IPv6 peers in the context-active VRF.
  ```
switch# clear ipv6 bgp errors
  ! Clearing all IPv4 and IPv6 peering sessions
  switch#
  ```
clear ipv6 bgp neighbor *

The **clear ipv6 bgp neighbor** * command clears BGP neighbors belonging to the IPv6 transport address family. To clear BGP neighbors in the IPv4 transport address family, use the **clear ip bgp neighbor** * command.

**Command Mode**

Privileged EXEC

**Command Syntax**

```
clear ipv6 bgp neighbor * [VRF_INSTANCE]
```

**Parameters**

- **VRF_INSTANCE** specifies VRF instance for which IPv6 transport address family BGP neighbors will be cleared. Options include:
  - `<no parameter>` clears IPv6 BGP neighbors in the context-active VRF.
  - `vrf vrf_name` clears IPv6 BGP neighbors in the specified VRF.
  - `vrf all` clears IPv6 BGP neighbors in the all VRFs.
  - `vrf default` clears IPv6 BGP neighbors in the default VRF.

**Examples**

- This command clears all IPv6 BGP neighbors in the context-active VRF.
  ```
  switch#clear ipv6 bgp neighbor *
  switch#
  ```

- This command clears all IPv6 BGP neighbors in VRF “purple.”
  ```
  switch#clear ipv6 bgp neighbor * vrf purple
  switch#
  ```
**distance bgp**

The **distance bgp** command assigns an administrative distance to routes that the switch learns through BGP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from 1 to 255; lower distance values correspond to higher reliability. BGP routing tables do not include routes with a distance of 255.

The distance command assigns distance values to external, internal, and local BGP routes:

- **external**: Best-path routes learned from a neighbor external to the autonomous system. Default distance is 200.
- **internal**: Internal routes are routes learned from a BGP entity within the same autonomous system. Default distance is 200.
- **local**: Local routes are networks listed with a network router configuration command for that router or for networks that are redistributed from another process. Default distance is 200.

The **no distance bgp** and **default distance bgp** commands restore the default administrative distances by removing the **distance bgp** command from **running-config**.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
distance bgp external_dist [INTERNAL_LOCAL]
no distance bgp
default distance bgp
```

**Parameters**

- **external_dist**  distance assigned to external routes. Values range from 1 to 255.
- **INTERNAL_LOCAL** distance assigned to internal and local routes. Values for both routes range from 1 to 255. Options include:
  - `<no parameter>`  **external_dist** value is assigned to internal and local routes.
  - **internal_dist**  **local_dist**  values assigned to internal (**internal_dist**) and local (**local_dist**) routes.

**Example**

- This command assigns an administrative distance of 150 to external routes, 200 to internal, and 150 to local routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#distance bgp 150 200 150
switch(config-router-bgp)#
```
dynamic peer max

The dynamic peer max command limits the number of dynamic BGP peers allowed on the switch. The no dynamic peer max and default dynamic peer max commands restore the default limit of dynamic BGP peers by removing the dynamic peer max command from running-config.

Command Mode
   Router-BGP Configuration

Command Syntax
   dynamic peer max maximum
   no dynamic peer max
   default dynamic peer max

Parameters
   • maximum  the maximum number of dynamic BGP peers to be allowed on the switch. Values range from 1 to 1000; default value is 100.

Example
   • This command sets the maximum number of dynamic BGP peers allowed on the switch to 200.
      switch(config)#router bgp 1
      switch(config-router-bgp)#dynamic peer max 200
      switch(config-router-bgp)#
**graceful-restart stalepath-time**

The `graceful-restart stalepath-time` command specifies the maximum time that stale routes from a restarting BGP neighbor will be retained after a BGP session is re-established with that peer.

The `no graceful-restart stalepath-time` and `default graceful-restart stalepath-time` commands restore the default value of 300 seconds by deleting the `graceful-restart stalepath-time` statement from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
graceful-restart stalepath-time interval
no graceful-restart stalepath-time
default graceful-restart stalepath-time
```

**Parameters**

- `interval` Maximum period (in seconds) that stale routes from a restarting BGP neighbor will be retained after the BGP session is re-established. Value ranges from 1 to 3600 (60 minutes). Default is 300.

**Example**

- These commands configure the stale path retention interval to 15 minutes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#graceful-restart stalepath-time 900
switch(config-router-bgp)#
```
graceful-restart-helper

The **graceful-restart helper** command enables BGP graceful restart helper mode on the switch for all BGP neighbors. When graceful restart helper mode is enabled, the switch will retain routes from neighbors which are capable of graceful restart while those neighbors are restarting BGP. Graceful restart is enabled by default. To configure graceful restart helper mode for a specific neighbor or peer group, use the `neighbor graceful-restart-helper` command. Individual neighbor configuration takes precedence over the global configuration.

The **no graceful-restart helper** command disables graceful restart helper mode on the switch. The **default graceful-restart helper** command enables graceful restart helper mode by removing the corresponding **no graceful-restart helper** command from `running-config`.

**Command Mode**

- Router-BGP Configuration

**Command Syntax**

```
graceful-restart helper
no graceful-restart helper
default graceful-restart helper
```

**Example**

- These commands disable graceful restart helper mode on the switch.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no graceful-restart-helper
switch(config-router-bgp)#
```
**ip as-path access-list**

The **ip as-path access-list** command creates an access list to filter BGP route updates. If access list **list_name** does not exist, this command creates it. If it already exists, this command appends statements to the list.

The **no ip as-path access-list** and **default ip as-path access-list** commands delete the named access list.

**Command Mode**
Global Configuration

**Command Syntax**

```
ip as-path access-list list_name FILTER_TYPE regex ORIGIN
no ip as-path access-list list_name
default ip as-path access-list list_name
```

**Parameters**
- **list_name** the name of the AS path access list.
- **FILTER_TYPE** access resolution of the specified AS path. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- **regex** a regular expression describing the AS path being filtered. Regular expressions are pattern matching strings that are composed of text characters and operators.
- **ORIGIN** the origin of the path information. Values include:
  - **<no parameter>** sets the origin to **any**.
  - **any** any BGP origin.
  - **egp** EGP origin.
  - **igp** IGP origin.
  - **incomplete** incomplete origin.

**Example**
- These commands create an AS path access list named “list1” which allows all BGP routes except those originating in AS 3.

```
switch(config)#ip as-path access-list list1 deny _3$
switch(config)#ip as-path access-list list1 permit .*
switch(config)#
```
ip as-path regex-mode

The `ip as-path regex-mode` command specifies how the switch will evaluate regular expressions describing AS paths in ACLs. When the regex mode is set to `asn`, AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match. When it is set to `string`, AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match. The default mode is `asn`.

For example, `asn` mode will return “false” and `string` mode will return “true” when searching for “10” in an AS path of “100 200”.

The `no ip as-path regex-mode` and `default ip as-path regex-mode` commands restore the regex mode to `asn` by removing the `ip as-path regex-mode` command from `running-config`.

**Command Mode**

- **Global Configuration**

**Command Syntax**

```
ip as-path regex-mode MODE_SETTING
no ip as-path regex-mode
default ip as-path regex-mode
```

**Parameters**

- **MODE_SETTING**  Specifies how regular expressions describing AS paths in AS path ACLs will be evaluated. Options include:
  - `asn`  AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match.
  - `string`  AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match.

**Example**

- This command sets the regex mode to `string`.

  ```
switch(config)#ip as-path regex-mode string
switch(config)#
```
**ip community-list**

The `ip community-list` command creates and configures a BGP access list based on BGP communities.

The `no ip community-list` and `default ip community-list` commands delete the specified community list by removing the corresponding `ip community-list` command from the running configuration.

**Command Mode**
Global Configuration

**Command Syntax**

```plaintext
ip community-list listname [permit | deny] [GSHUT | aa:nn | internet | local-as | no-advertise | no-export | number]
no ip community-list listname
default ip community-list listname
```

**Parameters**
- `listname` name of the community list. Valid input is text.
- `permit` permits access to the specified community.
- `deny` denies access to the specified community.
- `GSHUT` well-known graceful shutdown community.
- `aa:nn` AA is 65535 and NN specifies the community number (0-65535) within the AS.
- `internet` advertises route to the Internet community.
- `local-as` advertises route only to local peers.
- `no-advertise` does not advertise route to any peer.
- `no-export` advertises route only within BGP AS boundary.
- `number` community number. Value ranges from 0 to 4294967040.

**Related Commands**
- `route-map`
- `match (route-map)`
- `show ip community-list`
- `show ip extcommunity-list`

**Guideline**

EOS does not support disabling the process of graceful shutdown community.

**Examples**
- This command creates a BGP community list (named list_9) that does not match members of route maps configured as AS-network number 100:250.

  ```plaintext
  switch(config)#ip community-list list_9 deny 100:250
  switch(config)#
  ```
This command creates a BGP community list that permits the graceful shutdown community, and uses it in a route map to permit routes with that community.

```
switch(config)#ip community-list gshut_list permit GSHUT
switch(config)#route-map map1
switch(config-route-map-map1)#match community gshut_list
switch(config-route-map-map1)#exit
switch(config)#show route-map map1
route-map map1 permit 10
  Description:
  Match clauses:
    match community gshut_list
SubRouteMap:
Set clauses:
  switch(config)#
```
The `ip community-list regexp` command creates and configures a BGP access list based on BGP communities. A BGP community access list filters prefixes based on their BGP communities. The command uses regular expressions to identify the communities specified by the list. To create a community list by explicitly specifying one or more communities, use the `ip community-list` command.

The `no ip community-list regexp` and `default ip community-list regexp` commands delete the specified community list. To delete a specific community-list entry, specify the entry in the `no ip community-list regexp` command.

**Command Mode**
Global Configuration

**Command Syntax**

```
ip community-list regexp listname FILTER_TYPE R_EXP
no ip community-list regexp listname FILTER_TYPE R_EXP
default ip community-list regexp listname
```

**Parameters**

- `listname` name of the community list. Valid input is text.
- `FILTER_TYPE` access resolution of the specified community. Options include:
  - `permit` access is permitted.
  - `deny` access is denied.
- `R_EXP` list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- `entry` specifies a single entry to be removed from the list and leaves the rest of the list intact. Valid input is text. If no entry is specified, the `no` form of the command removes the entire list.

**Related Commands**

- `route-map`
- `match (route-map)`
- `show ip community-list`
- `show ip extcommunity-list`

**Example**

- This command creates a BGP community list that permits routes from networks 20-24 and 30-34 in autonomous system 10.
  
  ```
  switch(config)#ip community-list regexp list_2 permit 10:[2-3][0-4]_
  switch(config)#
  ```

- This command removes the above statement from the community list named “list_2,” leaving any other statements in the list intact.
  
  ```
  switch(config)#no ip community-list regexp list_2 permit 10:[2-3][0-4]_
  switch(config)#
  ```

- This command deletes the community list named “list_2” entirely.
  
  ```
  switch(config)#no ip community-list regexp list_2
  switch(config)#
  ```
ip extcommunity-list

The `ip extcommunity-list` command creates an extended community list to filter VRF routes or for link bandwidth (LBW) advertisement.

The following extcommunity-list types are supported:

- **Route Target (rt)** identifies sites that may receive appropriately tagged routes.
- **Site of Origin (soo)** identifies sites where the switch learned the route.
- **Link Bandwidth (lbw)** advertises BGP link bandwidth.

The `no ip extcommunity-list` and `default ip extcommunity-list` commands delete the specified extended community list by removing the corresponding `ip extcommunity-list` statement from `running-config`.

**Command Mode**

Global Configuration

**Command Syntax**

```
ip extcommunity-list listname FILTER_TYPE COMM_1 [COMM_2...COMM_n]
no ip extcommunity-list listname
default ip extcommunity-list listname
```

**Parameters**

- **listname** name of the extended community list. Valid input is text.
- **FILTER_TYPE** access resolution of the specified extended community list. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- **COMM_x** extended community attribute. Options include:
  - **rt aa:nn** route target, as specified by autonomous system:network number
  - **rt ip_addr:nn** route target, as specified by ip address:network number
  - **soo aa:nn** Site of Origin, as specified by autonomous system:network number
  - **soo ip_addr:nn** site of origin, as specified by ip address:network number
  - **lbw** link bandwidth in bits per second

**Related Commands**

- `route-map`
- `match (route-map)`
- `show ip community-list`
- `show ip extcommunity-list`

**Example**

- This command creates a BGP extended community list that denies routes from route target 100:250.

  ```
  switch(config)#ip extcommunity-list list_9 deny rt 100:250
  switch(config)#
  ```
The `ip extcommunity-list regexp` command creates an extended community list to filter VRF routes or for link bandwidth (LBW) advertisement. The command uses regular expressions to define the extended communities specified by the list. To specify particular values, use the `ip extcommunity-list` command.

The following extcommunity-list types are supported:

- **Route Target (rt)** identifies sites that may receive appropriately tagged routes.
- **Site of Origin (soo)** identifies sites where the switch learned the route.
- **Link Bandwidth (lbw)** advertises BGP link bandwidth.

The `no ip extcommunity-list regexp` and `default ip extcommunity-list regexp` commands delete the specified extended community list by removing the corresponding `ip extcommunity-list regexp` statement from `running-config`.

**Command Mode**

Global Configuration

**Command Syntax**

```
ip extcommunity-list regexp listname FILTER_TYPE R_EXP
no ip extcommunity-list regexp listname FILTER_TYPE R_EXP
default ip extcommunity-list regexp listname
```

**Parameters**

- **listname** name of the extended community list. Valid input is text.
- **FILTER_TYPE** access resolution of the specified extended community list. Options include:
  - `permit` access is permitted.
  - `deny` access is denied.
- **R_EXP** list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
  - Expressions beginning with `RT:` match the route target extended community attribute option.
  - Expressions beginning with `SoO:` match the site of origin extended community attribute option.

  `RT:` and `SoO:` are case sensitive.

**Related Commands**

- `route-map`
- `match (route-map)`
- `show ip community-list`
- `show ip extcommunity-list`

**Example**

- This command creates a BGP extended community list that denies routes from route target networks 20-24 and 30-34 in autonomous system 10.

```
switch(config)#ip extcommunity-list regexp list_1 deny RT:10:[2-3][0-4]_
switch(config)#
```
ip large-community-list regexp

The ip large-community-list regexp command creates and configures a BGP access list based on BGP large communities. A BGP large-community access list filters prefixes based on their BGP large community values. The command uses regular expressions to match large communities. Multiple large-community lists with the same name may be specified. To create a large-community list by explicitly specifying one or more communities, use the ip large-community-list command.

Large-communities are represented as follows: [ASN]:local-part1:local-part2.

The no ip large-community-list regexp and default ip large-community-list regexp commands delete the specified large community list. To delete a specific community-list entry, specify the entry in the no ip large-community-list regexp command.

Command Mode
Global Configuration

Command Syntax

```
ip large-community-list regexp listname {deny | permit} R_EXP
no ip large-community-list regexp listname {deny | permit} R_EXP
default ip large-community-list regexp listname {deny | permit}
```

Parameters
- **listname** name of the community list. Valid input is text.
- **deny** access is denied
- **permit** access is permitted
- **R_EXP** list of communities that are formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.

Example
- This command creates a BGP large community list that permits routes from autonomous system 10 with local-part1 value of 20-24 or 30-34.
  ```
  switch(config)#ip large-community-list regexp list_2 permit 10:[2-3][0-4]:_
  switch(config)#
  ```
- This command removes the above statement from the large community list named “list_2,” leaving any other statements in the list intact.
  ```
  switch(config)#no ip large-community-list regexp list_2 permit 10:[2-3][0-4]_
  switch(config)#
  ```
- This command deletes the large community list named “list_2” entirely.
  ```
  switch(config)#no ip large-community-list regexp list_2
  switch(config)#
  ```
maximum paths (BGP)

The **maximum-paths** command controls the maximum number of parallel eBGP routes that the switch supports. The default maximum is one route. The command provides an ECMP (equal cost multiple paths) parameter that controls the number of equal-cost paths that the switch stores in the routing table for each route.

For paths to be considered equal, they must have the same weight, local preference, AS-path length, and origin. To require that they also have the same multi-exit discriminator (MED) value, use the **bgp always-compare-med** command. To require that their AS paths have the same contents, use the **no bgp bestpath as-path multipath-relax** command.

The **no maximum-paths** and **default maximum-paths** commands restore the default values of the maximum number of parallel routes and the maximum number of ECMP paths by removing the corresponding command from **running-config**.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
maximum-paths paths [ecmp ecmp_paths]
no maximum-paths
default maximum-paths
```

**Parameters**

- **paths** maximum number of parallel routes. Default value is 1. Value must be less than or equal to the maximum number of ECMP paths.
- **ecmp_paths** maximum number of ECMP paths for each route. Default is maximum value.

Value for each parameter ranges from 1 to the number of interfaces available per ECMP group, which is platform-dependent.

- Arad: Value ranges from 1 to 128. Default value is 128.
- FM6000: Value ranges from 1 to 32. Default value is 32.
- PetraA: Value ranges from 1 to 16. Default value is 16.
- Trident: Value ranges from 1 to 32. Default value is 32.
- Trident-II: Value ranges from 1 to 128. Default value is 128.

**Examples**

- This command configures the maximum number of BGP parallel paths to 12 without changing the default ECMP value.

```
switch(config)#router bgp 1
switch(config-router-bgp)#maximum-paths 12
```

- This command configures the maximum number of eBGP parallel routes to 2, with a maximum of 4 equal cost (ECMP) paths for each route.

```
switch(config)#router bgp 1
switch(config-router-bgp)#maximum-paths 2 ecmp 4
```
match as-range

The **match as-range** command defines the match statement for the peer-filter, based on the match statement the peer-filter accept or reject the incoming peer request. The match statement includes a sequence number, AS number range and a match condition to accept or reject a peer by comparing its remote AS number to the specified range. A peer filter can consist of a single match statement or multiple match statements. The match statement for the peer filter is configured under `peer-filter` configuration mode.

The **no match as-range** or **default match as-range** command deletes the peer-filter condition from the group from the running configuration.

**Command Mode**
Peer-Filter Configuration

**Command Syntax**

```
[sequence_number] match as-range [as_number1] [- as_number2] result {accept | reject} group_name
no [sequence_number] match as-range [ASN1] [- ASN2] result {accept | reject} group_name
default [sequence_number] match as-range [ASN1] [- ASN2] result {accept | reject} group_name
```

**Parameters**
- `sequence_number` number ranges from 0 to 65535.
- `group_name` name of the peer filter group.
- `as_number` the autonomous system number, ranges from 1 to 4294967295.

**Example**

- These commands define a peer filter that accepts any AS number.
  ```
  switch(config)#peer-filter group1
  switch(config-peer-filter-group1)#10 match as-range 1-4294967295 result accept
  ```

- These commands define a peer filter that accepts any AS number within 65000 and 65100 (inclusive) except 65008 and 65009.
  ```
  switch(config)#peer-filter group2
  switch(config-peer-filter-group2)#10 match as-range 65008-65009 result reject
  switch(config-peer-filter-group2)#20 match as-range 65000-65100 result accept
  ```

- These commands define a peer filter that accepts 3 specific remote AS numbers.
  ```
  switch(config)#peer-filter group3
  switch(config-peer-filter-group3)#10 match as-range 65003 result accept
  switch(config-peer-filter-group3)#20 match as-range 65007 result accept
  switch(config-peer-filter-group3)#30 match as-range 65009 result accept
  ```
neighbor activate

The **neighbor activate** command defines the configuration mode address family activation state of a specified address that is configured as a BGP neighbor. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The **bgp default** command configures the default address family activation state of addresses configured as BGP neighbors. The **neighbor activate** and **no neighbor activate** commands override the neighbor’s default activation state within the configuration mode address family.

- **neighbor activate**: the specified address is active in the address family.
- **no neighbor activate**: the specified address is not active in the address family.

The **default neighbor activate** command removes the corresponding **neighbor activate** or **no neighbor activate** command from **running-config**, restoring the default address family activation state for the specified neighbor address.

**Command Mode**

Router-BGP Address-Family Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID activate
no neighbor NEIGHBOR_ID activate
default neighbor NEIGHBOR_ID activate
```

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - **ipv4_addr** neighbor’s IPv4 address.
  - **ipv6_addr** neighbor’s IPv6 address.
  - **group_name** peer group name.

**Limitations**

The switch supports the advertisement of networks with IPv6 prefixes to IPv4 transport neighbors. The switch does not support the advertisement of networks with IPv4 prefixes to IPv6 transport neighbors.

**Example**

- The two neighbor activation commands activate the advertising of specified neighbors during IPv4 peering sessions. The **show active** command displays the result of the previous commands.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no address-family ipv4
switch(config-router-bgp-af)#neighbor 172.41.18.15 activate
switch(config-router-bgp-af)#neighbor 172.49.22.6 activate
switch(config-router-bgp-af)#no neighbor 172.15.21.18 activate
switch(config-router-bgp-af)#show active
  address-family ipv4
    no neighbor 172.15.21.18 activate
    neighbor 172.49.22.6 activate
    neighbor 172.41.18.15 activate
switch(config-router-bgp-af)##exit
switch(config-router-bgp)#
```
neighbor allowas-in

The neighbor allowas-in command configures the switch to permit the advertisement of prefixes containing duplicate autonomous switch numbers (ASNs). This command programs the switch to ignore its ASN in the AS path of routes and allow them into the routing domain. This function is disabled by default.

The no neighbor allowas-in command applies the system default configuration.

The default neighbor allowas-in command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The no neighbor command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor NEIGHBOR_ID allowas-in [asn_quantity]
no neighbor NEIGHBOR_ID allowas-in
default neighbor NEIGHBOR_ID allowas-in

Parameters

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - ipv4_addr neighbor's IPv4 address.
  - ipv6_addr neighbor's IPv6 address.
  - group_name peer group name.
- **asn_quantity** Number of switches (ASN) allowed in path. Values range from 1 to 10. Default is 3.

Example

- This command activates the allowas-in function for the neighbor at 192.168.1.30.

  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 192.168.1.30 allowas-in
  switch(config-router-bgp)#
neighbor auto-local-addr

The `neighbor auto-local-addr` command configures the switch to automatically determine the local address to be used for the non-transport address family in NLRIIs sent to the specified neighbor or peer group. This allows IPv4 NLRIIs to be carried over IPv6 transport, or IPv6 NLRIIs to be carried over IPv4 transport.

The `no neighbor auto-local-addr` command applies the system default configuration.

The `default neighbor auto-local-addr` command applies the system default configuration for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

**Note**

While this feature works well in eBGP deployments in which the pairing routers are directly connected and have matching IP address configurations, multi-hop eBGP or iBGP deployments may require manual local address configuration.

To explicitly configure a local address for the non-transport address family for a specific neighbor or peer group, use the `neighbor local-v4-addr` command for IPv6 neighbors, or the `neighbor local-v6-addr` for IPv4 neighbors.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID auto-local-addr
no neighbor NEIGHBOR_ID auto-local-addr
default neighbor NEIGHBOR_ID auto-local-addr
```

**Parameters**

- **NEIGHBOR_ID**  IP address or peer group name. Values include:
  - `neighbor_addr` neighbor’s IP address.
  - `group_name` peer group name.

**Example**

- For the IPv6 neighbor at 2001:0DB8:c2a4:1761::2, these commands configure the switch to automatically determine the IPv4 NLRI value to be sent during peering sessions.

  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 2001:0DB8:c2a4:1761::2 auto-local-addr
  switch(config-router-bgp)#
neighbor default-originate

The `neighbor default-originate` command advertises a default route to a BGP neighbor or peer group. This default route overrides the default route advertised by any other means to the specified neighbor or peer group. However, the update generated by `neighbor default-originate` is not processed by neighbor route map out policies.

If a route map is specified in this command, its set clauses are used to modify attributes of the exported default route, but its match clauses are not used to conditionally advertise the route. The default route is always advertised to the specified neighbor.

The `no neighbor default-originate` command applies the system default configuration.

The `default neighbor default-originate` command applies the system default configuration for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

**Command Mode**
- Router-BGP Configuration
- Router-BGP Address-Family Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID default-originate [MAP]
no neighbor NEIGHBOR_ID default-originate
default neighbor NEIGHBOR_ID default-originate
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- `MAP` specifies route map that modifies attributes of the exported default route. Options include:
  - `<no parameter>` attributes are not modified by a route map.
  - `route-map map_name` attributes set by specified route map are assigned to the exported default route.

**Example**

- These commands advertise a default route to the BGP neighbor at 192.168.14.5.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 192.168.14.5 default-originate
  switch(config-router-bgp)#
  ```
neighbor description

The **neighbor description** command associates descriptive text with the specified peer or peer group.

The **no neighbor description** command removes the text association from the specified peer or peer group.

The **default neighbor description** command removes the text association from the specified peer for individual neighbors, and applies the peer group's description to neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

**Command Mode**
- Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID description description_string
no neighbor NEIGHBOR_ID description
default neighbor NEIGHBOR_ID description
```

**Parameters**

- **NEIGHBOR_ID**  
  IP address or peer group name. Options include:
  - `ipv4_addr`  
    neighbor's IPv4 address.
  - `ipv6_addr`  
    neighbor's IPv6 address.
  - `group_name`  
    peer group name.

- **description_string**  
  text string to be associated with the neighbor or peer group.

**Example**

- This command associates the string PEER_1 with the peer located at 192.168.1.30.

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 192.168.1.30 description PEER_1
  switch(config-router-bgp)#
  ```
neighbor ebgp-multihop

The **neighbor ebgp-multihop** command programs the switch to accept and attempt BGP connections to the external peers residing on networks not directly connected to the switch. The command does not establish the multihop if the only route to the peer is the default route (0.0.0.0).

The **no neighbor ebgp-multihop** command applies the system default configuration.

The **default neighbor ebgp-multihop** command applies the system default configuration for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID ebgp-multihop [hop_number]
no neighbor NEIGHBOR_ID ebgp-multihop
default neighbor NEIGHBOR_ID ebgp-multihop
```

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor’s IPv4 address.
  - *ipv6_addr* neighbor’s IPv6 address.
  - *group_name* peer group name.
- **hop_number** time-to-live (hops). Values range from 1 to 255. Default value is 255.

**Example**

- This command programs the switch to accept and attempt BGP connections to the external peer located at 192.168.1.30, setting the hop limit to 32.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 ebgp-multihop 32
switch(config-router-bgp)#
```
neighbor enforce-first-as

The `neighbor enforce-first-as` command causes a forced comparison of the first autonomous system (AS) in the AS path of eBGP routes received from a specified BGP peer or peer group to the configured remote external peer autonomous system number (ASN). Updates from the specified eBGP peers that do not include an ASN as first AS path (in the AS_PATH attribute) are discarded.

This behavior is enabled globally by default upon BGP configuration, and disabled for the specified neighbor or peer group by the `no` form of the command. To configure enforce-first-as globally, use the `bgp enforce-first-as` command.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID enforce-first-as
default neighbor NEIGHBOR_ID enforce-first-as
no neighbor NEIGHBOR_ID enforce-first-as
```

**Parameters**

- **NEIGHBOR_ID**  IP address or peer group name. Values include:
  - `ipv4_addr`  neighbor’s IPv4 address.
  - `ipv6_addr`  neighbor’s IPv6 address.
  - `group_name`  peer group name.

**Example**

- This command disables BGP enforce-first-as for the neighbors in peer group “region-3”.

  ```
switch(config-router-bgp)#no neighbor region-3 enforce-first-as
switch(config-router-bgp)#
```
neighbor export-localpref

The `neighbor export-localpref` command determines the LOCAL_PREF value that is sent in BGP UPDATE packets to the specified peer or peer group. This command has no effect on external peers.

The `no neighbor export-localpref` command resets the LOCAL_PREF value to the system default of 100 in packets sent to the specified peer or peer group.

The `default neighbor export-localpref` command resets the LOCAL_PREF value to the system default of 100 for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address or the specified peer group.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID export-localpref preference
no neighbor NEIGHBOR_ID export-localpref
default neighbor NEIGHBOR_ID export-localpref
```

**Parameters**

- **NEIGHBOR_ID**  
  IP address or peer group name. Values include:
  - `ipv4_addr`  
    neighbor’s IPv4 address.
  - `ipv6_addr`  
    neighbor’s IPv6 address.
  - `group_name`  
    peer group name.
- **preference**  
  preference value. Values range from 0 to 4294967295.

**Example**

- This command configures the switch to fill the LOCAL_PREF field with 200 in UPDATE packets that it sends to the peer located at 10.1.1.45.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```
neighbor graceful-restart-helper

The `neighbor graceful-restart helper` command enables BGP graceful restart helper mode for the specified BGP neighbor or peer group. When graceful restart helper mode is enabled, the switch will retain routes from neighbors which are capable of graceful restart while those neighbors are restarting BGP. Graceful restart is enabled by default for all BGP neighbors. To configure graceful restart helper mode for all BGP neighbors, use the `graceful-restart-helper` command. Individual neighbor configuration takes precedence over the global configuration.

The `no neighbor graceful-restart helper` command disables graceful restart helper mode for the specified BGP neighbor or peer group. The `default neighbor graceful-restart helper` command enables graceful restart helper mode for the specified BGP neighbor or peer group by removing the corresponding `no neighbor graceful-restart helper` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID graceful-restart helper
no neighbor NEIGHBOR_ID graceful-restart helper
default neighbor NEIGHBOR_ID graceful-restart helper
```

**Parameters**

- `NEIGHBOR_ID`  IP address or peer group name. Values include:
  - `ipv4_addr`  neighbor's IPv4 address.
  - `ipv6_addr`  neighbor's IPv6 address.
  - `group_name`  peer group name.

**Example**

- These commands disable graceful restart helper mode for the neighbor at 192.168.12.1.
  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#no neighbor 192.168.12.1 graceful-restart-helper
  switch(config-router-bgp)#
  ```
neighbor import-localpref

The `neighbor import-localpref` command determines the local preference assigned to routes received from the specified external peer or peer group. This command has no effect on routes received from internal peers.

The `no neighbor import-localpref` command resets the local preference to the default of 100 for routes received from the specified peer or peer group.

The `default neighbor import-localpref` command resets the local preference to the default of 100 for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID import-localpref preference
no neighbor NEIGHBOR_ID import-localpref
default neighbor NEIGHBOR_ID import-localpref
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- `preference` preference value. Values range from 0 to 4294967295.

**Example**

- This command configures the switch to assign a local preference of 50 to routes received from the peer located at 192.168.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 import-localpref 50
switch(config-router-bgp)#
```
neighbor local-as

The `neighbor local-as` command enables AS_PATH attribute modification for received eBGP routes, allowing the switch to appear as a member of a different AS to external peers. Arista switches do not prepend the local AS number to routes received from the eBGP neighbor; currently, we implement the command only as `neighbor local-as no-prepend replace-as`.

The `no neighbor local-as` command disables AS_PATH modification for the specified peer or peer group. The `default neighbor local-as` command disables AS_PATH modification for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID local-as as_id no-prepend replace-as
no neighbor NEIGHBOR_ID local-as
default neighbor NEIGHBOR_ID local-as
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor's IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- `as_id` AS number that is sent in outbound routing updates in place of the actual AS of the switch. Values range from 1 to 4294967295.

This parameter cannot be set to the switch’s AS number or to any AS number in the peer’s network.

**Example**

- For the neighbor at 10.13.64.1, this command removes AS 300 from outbound routing updates and replaces it with AS 600.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.13.64.1 local-as 600 no-prepend replace-as
switch(config-router-bgp)#
```
neighbor local-v4-addr

The `neighbor local-v4-addr` command specifies the next-hop value that the switch sends as the IPv4 NLRI value to neighbors with whom IPv6 transport peering is established.

The `no neighbor local-v4-addr` command applies the system default configuration.

The `default neighbor local-v4-addr` command applies the system default configuration for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

To configure the switch to automatically determine the IPv4 address to be sent as the next-hop in IPv4 NLRIs to an IPv6 neighbor, use the `neighbor auto-local-addr` command.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID local-v4-addr ipv4_local
no neighbor NEIGHBOR_ID local-v4-addr
default neighbor NEIGHBOR_ID local-v4-addr
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
  - `ipv4_local` Next hop address.

**Example**

- For the neighbor at 2001:0DB8:c2a4:1761::2, these commands specify an IPv4 NLRI value of 10.7.5.11 to be sent during IPv6 transport peering sessions.
  
  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 2001:0DB8:c2a4:1761::2 local-v4-addr 10.7.5.11
  switch(config-router-bgp)#


neighbor local-v6-addr

The neighbor local-v6-addr command specifies the next-hop value that the switch sends as the IPv6 NLRI value to neighbors with whom IPv4 transport peering is established.

In IPv6 peering sessions, the switch sends the global IPv6 address of the interface that is used to transmit BGP updates.

The no neighbor local-v6-addr command applies the system default configuration.

The default neighbor local-v6-addr command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-v6-addr ipv6_local
no neighbor NEIGHBOR_ID local-v6-addr
default neighbor NEIGHBOR_ID local-v6-addr
```

Parameters

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor's IPv4 address.
  - `group_name` peer group name.

Example

- For the neighbor at 10.7.5.11, these commands specify an IPv6 NLRI value that is sent during IPv4 transport peering sessions.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.7.5.11 local-v6-addr 2001:0DB8:c2a4:1761::2
switch(config-router-bgp)#show active
router bgp 1
  bgp log-neighbor-changes
  bgp default ipv6-unicast
  neighbor 10.7.5.11 local-v6-addr 2001:0DB8:c2a4:1761::2
switch(config-router-bgp)#
```
neighbor maximum-routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor and defines an action when the limit is exceeded. The default value is 12,000. To remove the maximum routes limit, select a limit of zero.

When the number of routes received from a peer exceeds the limit, the switch generates an error message. This command can also configure the switch to disable peering with the neighbor. In this case, the neighbor state is reset only through a `clear ip bgp` command.

The **no neighbor maximum-routes** command applies the system default maximum-routes value of 12,000 for the specified peer.

The **default neighbor maximum-routes** command applies the system default value for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

**Command Mode**

- Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID maximum-routes quantity [ACTION]
no neighbor NEIGHBOR_ID maximum-routes
default neighbor NEIGHBOR_ID maximum-routes
```

**Parameters**

- **NEIGHBOR_ID**  IP address or peer group name. Values include:
  - `ipv4_addr`  neighbor’s IPv4 address.
  - `ipv6_addr`  neighbor’s IPv6 address.
  - `group_name`  peer group name.
- **quantity**  maximum number of routes. Values include:
  - 0  the switch does not define a route limit.
  - 1 to 4294967294  maximum number of routes.
- **ACTION**  switch action when the route limit is exceeded. Values include:
  - `<no parameter>`  peering is disabled and an error message is generated.
  - `warning-only`  peering is not disabled, but an error message is generated.

**Example**

- This command configures the switch to accept 15000 routes for the neighbor at 10.3.16.210. If the neighbor exceeds 15000 routes, the switch disables peering with the neighbor.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 110.3.16.210 maximum-routes 15000
switch(config-router-bgp)#
```
neighbor next-hop-peer

The `neighbor next-hop-peer` command configures the switch to list the peer address as the next hop in routes that it receives from the specified peer BGP-speaking neighbor or members of the specified peer group. This command overrides the next hop for all routes received from this neighbor or peer group.

The `no neighbor next-hop-peer` command applies the system default (no next-hop override) for the specified peer.

The `default neighbor next-hop-peer` command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address or the specified peer group.

**Command Mode**
Router-BGP Configuration

**Command Syntax**
```
neighbor NEIGHBOR_ID next-hop-peer
no neighbor NEIGHBOR_ID next-hop-peer
default neighbor NEIGHBOR_ID next-hop-peer
```

**Parameters**
- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor's IPv4 address.
  - `ipv6_addr` neighbor's IPv6 address.
  - `group_name` peer group name.

**Example**
- This command configures the peer address of 10.3.2.24 as the next hop for routes advertised to the switch from the peer BGP neighbor.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.3.2.24 next-hop-peer
switch(config-router-bgp)#
```
neighbor next-hop-self

The **neighbor next-hop-self** command configures the switch to list its address as the next hop in routes that it advertises to the specified BGP-speaking neighbor or neighbors in the specified peer group. This is used in networks where BGP neighbors do not directly access all other neighbors on the same subnet.

The **no neighbor next-hop-self** command applies the system default (no next-hop override) for the specified peer.

The **default neighbor next-hop-self** command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

**Command Mode**

   Router-BGP Configuration

**Command Syntax**

- `neighbor NEIGHBOR_ID next-hop-self`
- `no neighbor NEIGHBOR_ID next-hop-self`
- `default neighbor NEIGHBOR_ID next-hop-self`

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.

**Example**

- This command configures the switch as the next hop for the peer at 10.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.4.1.30 next-hop-self
switch(config-router-bgp)#
```
neighbor out-delay

The **neighbor out-delay** command sets the period of time that a route update for the specified neighbor must be in the routing table before the switch exports it to BGP. The out delay interval is used for bundling routing updates.

The **no neighbor out-delay** command applies the system default (out-delay value of zero) for the specified peer.

The **default neighbor out-delay** command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

**Command Mode**
Router-BGP Configuration

**Command Syntax**
neighbor **NEIGHBOR_ID** out-delay **delay_time**
no neighbor **NEIGHBOR_ID** out-delay
default neighbor **NEIGHBOR_ID** out-delay

**Parameters**
- **NEIGHBOR_ID**  IP address or peer group name. Values include:
  - ipv4_addr  neighbor’s IPv4 address.
  - ipv6_addr  neighbor’s IPv6 address.
  - group_name  peer group name.
- **delay_time**  the out delay period (seconds). Values range from 0 to 600. Default value is 0.

**Example**
- This command sets the out delay period to 5 seconds for the connection with the peer at 10.24.15.9.
  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 10.24.15.9 out-delay 5
  switch(config-router-bgp)#
neighbor passive

The **neighbor passive** command sets the TCP connection for the specified BGP neighbor or peer group to passive mode. When the peer’s transport connection mode is set to passive, it accepts TCP connections for BGP but does not initiate them.

The **no neighbor passive** command sets the specified BGP neighbor or peer group to active connection mode. BGP peers in active mode can both accept and initiate TCP connections for BGP. This is the default behavior.

The **default neighbor passive** command restores the default connection mode. The default mode is “active” for individual BGP peers, or the mode inherited from the peer group for peer group members.

**Command Mode**
- Router-BGP Configuration

**Command Syntax**
- `neighbor NEIGHBOR_ID passive`
- `no neighbor NEIGHBOR_ID passive`
- `default neighbor NEIGHBOR_ID passive`

**Parameters**
- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.

**Example**
- These commands configure the neighbor at IP address 10.2.2.14 to not initiate TCP connections for BGP peering.
  ```
  switch(config)#router bgp 300
  switch(config-router-bgp)#neighbor 10.2.2.14 passive
  switch(config-router-bgp)#
  ```
**neighbor password**

The `neighbor password` command enables authentication on a TCP connection with a BGP peer. The plain-text version of the password is a string, up to 8 bytes in length. Peers must use the same password to ensure proper communication.

*Running-config* displays the encrypted version of the password. The encryption scheme is not strong by cryptographic standards; encrypted passwords should be treated in the same manner as plain-text passwords.

The `no neighbor password` command applies the system default for the specified peer, removing the neighbor password from the configuration and disabling authentication with the specified peer.

The `default neighbor password` command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor password` and `default neighbor password` commands remove the neighbor password from the configuration, disabling authentication with the specified peer.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID password [ENCRYPT_LEVEL] key_text
no neighbor NEIGHBOR_ID password
default neighbor NEIGHBOR_ID password
```

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- **ENCRYPT_LEVEL** the encryption level of the `key_text` parameter. Values include:
  - `<no parameter>` indicates the `key_text` is in clear text.
  - `0` indicates `key_text` is in clear text. Equivalent to the `<no parameter>` case.
  - `7` indicates `key_text` is md5 encrypted.
- **key_text** the password.

**Example**

- This command specifies a password in clear text.
  
  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#neighbor 10.25.25.13 password 0 code123
  switch(config-router-bgp)#
  ```

  *Running-config* stores the password as an encrypted string.
neighbor peer group (create)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The `neighbor peer group (create)` command is used to create static BGP peer groups. Static peer groups are peer groups whose members are added manually. To assign BGP neighbors to a static peer group, use the `neighbor peer-group (neighbor assignment)` command. To create a dynamic peer group, use the `bgp listen range` command.

The `no neighbor peer group (create)` and `default neighbor peer group (create)` commands remove the specified static peer group from `running-config`. When a static peer group is deleted, the neighbors that were members of that peer group lose any configuration that was inherited from the peer group. The `no bgp listen range` command removes a dynamic peer group.

The `no neighbor` command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor group_name peer group
no neighbor group_name peer group
default neighbor group_name peer group
```

Parameters

- `group_name` peer group name.

Examples

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, apply a route map and adjust the configuration for one group member.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor bgpgroup1 peer group
switch(config-router-bgp)#neighbor 10.1.1.1 peer group bgpgroup1
switch(config-router-bgp)#neighbor 10.2.2.2 peer group bgpgroup1
switch(config-router-bgp)#neighbor 10.3.3.3 peer group bgpgroup1
switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#show active
router bgp 9
bgp log-neighbor-changes

neighbor bgpgroup1 peer-group
neighbor bgpgroup1 route-map corporate in
neighbor bgpgroup1 maximum-routes 12000
neighbor 10.1.1.1 peer-group bgpgroup1
neighbor 10.2.2.2 peer-group bgpgroup1
neighbor 10.3.3.3 peer-group bgpgroup1
neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```
This command removes peer group “bgpgroup1” from running-config. The group members remain, but all settings that group members inherited from the peer group are removed.

```bash
switch(config-router-bgp)#no neighbor bgpgroup1 peer-group
switch(config-router-bgp)#show active
router bgp 9
  bgp log-neighbor-changes
    neighbor 10.1.1.1 maximum-routes 12000
    neighbor 10.2.2.2 maximum-routes 12000
    neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```
neighbor peer-group (neighbor assignment)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer-group** (neighbor assignment) command is used to assign BGP neighbors to an existing static peer group. To create a static peer group, use the **neighbor peer group (create)** command. A neighbor can only belong to one peer group, so issuing this command for a neighbor that is already a member of another group will remove it from that group.

The **no neighbor peer-group** and **default neighbor peer-group** commands remove the specified neighbor from all peer groups. When a neighbor is removed from a peer group, the neighbor retains the configuration inherited from the peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ADDR peer-group group_name
no neighbor NEIGHBOR_ADDR peer-group
default neighbor NEIGHBOR_ADDR peer-group
```

**Parameters**

- **NEIGHBOR_ADDR** Address of a neighbor being added to peer group. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
- **group_name** peer group name.

**Examples**

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, and apply a route map.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor bgpgroup1 peer-group
  switch(config-router-bgp)#neighbor 10.1.1.1 peer-group bgpgroup1
  switch(config-router-bgp)#neighbor 10.2.2.2 peer-group bgpgroup1
  switch(config-router-bgp)#neighbor 10.3.3.3 peer-group bgpgroup1
  switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
  ```
- This command removes the neighbor at 1.1.1.1 from the peer group. All settings that neighbor 10.1.1.1 inherited from the peer group are maintained.
  ```
  switch(config-router-bgp)#no neighbor 10.1.1.1 peer-group
  ```
neighbor remote-as

The `neighbor remote-as` command configures the expected AS number for a neighbor (peer). This configuration is required to establish a neighbor connection. Internal neighbors have the same AS number; external neighbors have different AS numbers.

**Note**
To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the `router-id (BGP)` command.

The `no neighbor remote-as` command applies the system default for the specified peer or peer group. The `default neighbor remote-as` command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

**Command Mode**
Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID remote-as as_id
no neighbor NEIGHBOR_ID remote-as
default neighbor NEIGHBOR_ID remote-as
```

**Parameters**
- `NEIGHBOR_ID`  IP address or peer group name. Values include:
  - `ipv4_addr`  neighbor’s IPv4 address.
  - `ipv6_addr`  neighbor’s IPv6 address.
  - `group_name`  peer group name.
  - `as_id`  Autonomous system (AS) of the peer. Values range from 1 to 4294967295.

**Example**
- This command establishes a BGP connection with the router at 10.4.3.10 in AS 300.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.4.3.10 remote-as 300
switch(config-router-bgp)#
```
neighbor remove-private-as

The `neighbor remove-private-as` command removes private autonomous system numbers from outbound routing updates for external BGP (eBGP) neighbors. When the autonomous system path includes only private autonomous system numbers, the `REMOVAL` parameter specifies how the private autonomous system number is removed.

The `no neighbor remove-private-as` command applies the system default (preserves private AS numbers) for the specified peer.

The `default neighbor remove-private-as` command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```plaintext
neighbor NEIGHBOR_ID remove-private-as [REMOVAL]
no neighbor NEIGHBOR_ID remove-private-as
default neighbor NEIGHBOR_ID remove-private-as
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.

- `REMOVAL` Specifies removal of all private AS numbers when the AS path contains only private AS numbers. Values include:
  - `all` removes all private AS numbers from AS path in outbound updates.
  - `all replace-as` all private AS numbers in AS path are replaced with router’s local AS number.

**Note**

This command does not support a mix of public and private AS numbers.

**Example**

- These commands program the switch to remove all private AS numbers from outbound routing updates for the eBGP neighbor at 10.5.2.11 only if the AS path does not contain any public AS number.

  ```plaintext
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.5.2.11 remove-private-as
  switch(config-router-bgp)#
  ```

- This command replaces all private AS numbers in the AS path with the switch’s local AS number.

  ```plaintext
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.5.2.11 remove-private-as all replace-as
  switch(config-router-bgp)#
  ```
neighbor rib-in pre-policy retain

By default, inbound BGP routes that are filtered out by the import policy are still stored on the switch. Because all routes are retained, this allows policies to be changed without resetting BGP sessions. It also allows the switch to display all advertised routes when the `show ip bgp neighbor advertised-routes` command is issued.

The `no neighbor rib-in pre-policy retain` command configures the switch to discard information about routes received from the specified neighbor or group that fail the import policy.

The `neighbor rib-in pre-policy retain` command restores the system default behavior (retaining routes from the specified neighbor or group regardless of import policy).

The `default neighbor rib-in pre-policy retain` command applies the system default (retaining all routes) for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID rib-in pre-policy retain inbound [SCOPE]
no neighbor NEIGHBOR_ID rib-in pre-policy retain inbound
default neighbor NEIGHBOR_ID rib-in pre-policy retain inbound
```

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- **SCOPE** determines how routes including the switch’s AS number are handled. Values include:
  - `<no parameter>` routes including the switch’s AS number are discarded.
  - `all` routes including the switch’s AS number are retained.

**Example**

- This command configures the switch to discard information about routes from the neighbor at 10.5.2.23 which are filtered out by the switch’s import policies.

  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#no neighbor 10.5.2.23 rib-in pre-policy retain inbound
  switch(config-router-bgp)#
  ```
neighbor route-map (BGP)

The neighbor route-map command applies a route map to inbound or outbound BGP routes. When a route map is applied to outbound routes, the switch will advertise only routes matching at least one section of the route map. Only one outbound route map and one inbound route map can be applied to a given neighbor. A new route map applied to a neighbor will replace the previous route map.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is executed determines the scope of the command:

- In Router-BGP mode, the route map is applied to specified neighbor in all peering sessions where it is advertised.
- In Router-BGP-Address-Family mode, the route map is applied to the neighbors only in peering sessions corresponding to the configuration mode address family.

The no neighbor route-map command discontinues the application of the specified route map for the specified neighbor and direction. Removing a route map from one direction does not remove it from the other if it has been applied to both.

The default neighbor route-map command applies the system default (no route map) for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration
Router-BGP Address-Family Configuration

Command Syntax

```
neighbor NEIGHBOR_ID route-map map_name DIRECTION
no neighbor NEIGHBOR_ID route-map map_name DIRECTION
default neighbor NEIGHBOR_ID route-map map_name DIRECTION
```

Parameters

- **NEIGHBOR_ID**  
  IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- **map_name**  
  name of a route map.
- **DIRECTION**  
  routes to which the route map is applied. Options include:
  - `in` route map is applied to inbound routes.
  - `out` route map is applied to outbound routes.

Example

- This command applies a route map named `inner-map` to a BGP inbound route from 10.5.2.11

  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.5.2.11 route-map inner-map in
  switch(config-router-bgp)#
  ```
neighbor route-reflector-client

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers, but to prevent routing loops they must not re-advertise iBGP-learned routes within the AS. To ensure that all members of the AS share the same routing information, a fully meshed network topology (in which each member router of the AS is connected to every other member) can be used, but this topology can result in high volumes of iBGP messages when it is scaled. Instead, in larger networks one or more routers can be configured as route reflectors.

A route reflector is configured to re-advertise routes learned through iBGP to a group of BGP neighbors within the AS (its clients), eliminating the need for a fully meshed topology.

The neighbor route-reflector-client command configures the switch to act as a route reflector and configures the specified neighbor as one of its clients. Additional clients are specified by re-issuing the command.

The no neighbor route-reflector-client and default neighbor route-reflector-client commands disable route reflection by deleting the neighbor route-reflector-client command from running-config.

Command Mode
Router-BGP Configuration

Command Syntax
neighbor NEIGHBOR_ID route-reflector-client
no neighbor NEIGHBOR_ID route-reflector-client
default neighbor NEIGHBOR_ID route-reflector-client

Parameters
• **NEIGHBOR_ID** IP address of neighbor. Values include:
  • *ipv4_addr* neighbor’s IPv4 address.
  • *ipv6_addr* neighbor’s IPv6 address.
  • *group_name* peer group name.

Related Commands
• bgp client-to-client reflection

Example
• This command configures the switch as a route reflector and the neighbor at 10.5.2.1 as one of its clients.
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.5.2.11 route-reflector-client
  switch(config-router-bgp)#
neighbor send-community

The `neighbor send-community` command configures the switch to send community attributes to the specified BGP neighbor.

The `no neighbor send-community` command applies the system default (not sending community attributes) for the specified peer.

The `default neighbor send-community` command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID send-community
no neighbor NEIGHBOR_ID send-community
default neighbor NEIGHBOR_ID send-community
```

**Parameters**

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.

**Example**

This command configures the switch to send community attributes to the neighbor at address 10.5.2.23.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.23 send-community
switch(config-router-bgp)#
```
neighbor shutdown

The **neighbor shutdown** command disables the specified neighbor. Disabling a neighbor also terminates all of its active sessions and removes associated routing information.

The **no neighbor shutdown** command enables the specified peer.

The default neighbor shutdown command enables individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
neighbor NEIGHBOR_ID shutdown
no neighbor NEIGHBOR_ID shutdown
default neighbor NEIGHBOR_ID shutdown
```

**Parameters**

- **NEIGHBOR_ID** IP address or peer group name. Values include:
  - **ipv4_addr** neighbor’s IPv4 address.
  - **ipv6_addr** neighbor’s IPv6 address.
  - **group_name** peer group name.

**Example**

- This command disables the neighbor at 10.5.2.23.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.5.2.23 shutdown
  switch(config-router-bgp)#
  ```
neighbor timers

The `neighbor timers` command configures the BGP keepalive and hold times for a specified peer connection. The `timers bgp` command configures the times on all peer connections for which an individual command is not specified.

- Keepalive time is the period between the transmission of consecutive keepalive messages.
- Hold time is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The `no neighbor timers` command applies the system default for the specified peer or group (the timers specified by the `timers bgp` command).

The default neighbor timers command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID timers keep_alive hold_time  
no neighbor NEIGHBOR_ID timers  
default neighbor NEIGHBOR_ID timers
```

Parameters

- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- `keep_alive` keepalive period, in seconds. Values include
  - `0` keepalive messages are not sent
  - `1` to `3600` keepalive time (seconds).
- `hold_time` hold time. Values include
  - `0` peering is not disabled by timeout expiry; keepalive packets are not sent.
  - `3` to `7200` hold time (seconds).

Example

- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds for the connection with the peer at 10.24.15.9.

```bash
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.24.15.9 timers 30 90
switch(config-router-bgp)#
```
neighbor ttl maximum-hops

The **neighbor ttl maximum-hops** command configures the Generalized TTL Security Mechanism (GTSM) for the specified neighbor(s).

The **no neighbor ttl maximum-hops** command disables the GTSM configuration in the specified neighbor.

The **default neighbor ttl maximum-hops** command applies the system default configuration for individual neighbors; and applies the peer group’s setting for neighbors that are members of a peer group.

**Command-Mode**
Router-BGP Configuration

**Command Syntax**

```
neighbor neighbor_id ttl maximum-hops n
default neighbor neighbor_id ttl maximum-hops
no neighbor neighbor_id ttl maximum-hops
```

**Parameters**

- **neighbor_id**  IP address or peer group name. Values include:
  - **ipv4_addr**  neighbor's IPv4 address.
  - **ipv6_addr**  neighbor's IPv6 address.
  - **group_name**  peer group name.
- **n**  Maximum count of hops from a BGP peer. The value ranges from 0 to 254.

**Example**

- This command configures the TTL security for 10.20.20.30 with a maximum of 4 hops.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.20.20.30 ttl maximum-hops 4
switch(config-router-bgp)#
```
neighbor update-source

The neighbor update-source command specifies the interface that BGP sessions use for TCP connections. By default, BGP sessions use the neighbor’s closest interface (also known as the best local address).

The no neighbor update-source command applies the system default (using best local address for TCP connections) for the specified peer or group.

The default neighbor update-source command applies the system default for individual neighbors and applies the peer group’s setting for neighbors that are members of a peer group.

The no neighbor command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

neighbor NEIGHBOR_ID update-source INTERFACE
no neighbor NEIGHBOR_ID update-source
default neighbor NEIGHBOR_ID update-source

Parameters

- **NEIGHBOR_ID**  IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- **INTERFACE** Interface type and number. Options include:
  - `ethernet e_num` Ethernet interface specified by `e_num`.
  - `loopback l_num` loopback interface specified by `l_num`.
  - `management m_num` management interface specified by `m_num`.
  - `port-channel p_num` port channel interface specified by `p_num`.
  - `vlan v_num` VLAN interface specified by `v_num`.

Example

- This command configures the switch to use Ethernet interface 10 for TCP connections for the neighbor at 10.2.2.14.
  
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.2.2.14 update-source ethernet 10
  switch(config-router-bgp)#
neighbor weight

The `neighbor weight` command assigns a weight attribute value to paths from the specified neighbor. Weight is the first parameter that the BGP best-path selection algorithm considers. When multiple paths to a destination prefix exist, the best-path selection algorithm prefers the path with the highest weight. Other attributes are used only when all paths to the prefix have the same weight.

Weight values range from 0 to 65535 and are not propagated to other switches through route updates. The default weight for paths that the router originates is 32768; the default weight for routes received through BGP is 0.

A path’s BGP weight is also configurable through route maps. Weight values set through route map commands have precedence over neighbor weight command values.

The `no neighbor weight` command applies the system default (32768 for router-originated paths, 0 for routes received through BGP) for the specified peer or group.

The `default neighbor weight` command applies the system default for individual neighbors, and applies the peer group’s setting for neighbors that are members of a peer group.

The `no neighbor` command removes all configuration commands for the neighbor at the specified address.

Command Mode
- Router-BGP Configuration

Command Syntax
```
neighbor NEIGHBOR_ID weight weight_value
no neighbor NEIGHBOR_ID weight
default neighbor NEIGHBOR_ID weight
```

Parameters
- `NEIGHBOR_ID` IP address or peer group name. Values include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.
  - `group_name` peer group name.
- `weight_value` weight value. Values range from 1 to 65535.

Example
- This command specifies a weight of 4000 for all paths from the neighbor at 10.1.2.5
  
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#neighbor 10.1.2.5 weight 4000
  switch(config-router-bgp)#
  ```
network (BGP)

The `network` command specifies a network for advertisement through UPDATE packets to BGP peers. The configuration zeros the host portion of the specified network address; for example, 192.0.2.4/24 is stored as 192.0.2.0/24. A route map option is available for assigning attributes to the network.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is issued does not affect the command. The scope of the command depends on the specified network address:

- Commands with an IPv4 address are advertised to peers activated in the IPv4 address family.
- Commands with an IPv6 address are advertised to peers activated in the IPv6 address family.

The `no network` and `default network` commands remove the network from the routing table, preventing its advertisement.

**Command Mode**
- Router-BGP Configuration
- Router-BGP Address-Family Configuration

**Command Syntax**

```
network NET_ADDRESS [ROUTE_MAP]
no network NET_ADDRESS
default network NET_ADDRESS
```

**Parameters**

- **NET_ADDRESS**  IP address range. Entry options include:
  - `ipv4_subnet` IPv4 subnet (CIDR notation).
  - `ipv4_addr mask subnet` IPv4 subnet (address-mask notation).
  - `ipv6_prefix` neighbor’s IPv6 prefix (CIDR notation).
- **ROUTE_MAP** specifies route map that assigns attribute values to the network. Options include:
  - `<no parameter>` attributes are not assigned through a route map.
  - `route-map map_name` attributes listed by specified route map are assigned to the network.

**Example**

- This command enables BGP advertising for the network located at 10.1.2.5. The configuration stores the network as 10.1.2.5.
  
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#network 10.1.2.5/24
  switch(config-router-bgp)#
  ```
**no neighbor**

The `no neighbor` command removes all neighbor configuration commands for the specified neighbor. Commands removed by the `no neighbor` command include:

- `neighbor description`
- `neighbor ebgp-multihop`
- `neighbor export-localpref`
- `neighbor import-localpref`
- `neighbor local-as`
- `neighbor maximum-routes`
- `neighbor next-hop-peer`
- `neighbor next-hop-self`
- `neighbor out-delay`
- `neighbor password`
- `neighbor peer group (create)`
- `neighbor peer-group (neighbor assignment)`
- `neighbor remote-as`
- `neighbor remove-private-as`
- `neighbor route-map (BGP)`
- `neighbor route-reflector-client`
- `neighbor send-community`
- `neighbor timers`
- `neighbor update-source`

Neighbor settings can be removed individually; refer to the command description page of the desired command for details. Neighbor settings for a peer group must be removed individually.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
no neighbor NEIGHBOR_ID
default neighbor NEIGHBOR_ID
```

**Parameters**

- `NEIGHBOR_ID` IP address. Options include:
  - `ipv4_addr` neighbor’s IPv4 address.
  - `ipv6_addr` neighbor’s IPv6 address.

**Example**

- This command removes all neighbor configuration commands for the neighbor at 10.1.1.1.

  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#no neighbor 10.1.1.1
  switch(config-router-bgp)#
  ```
**peer-filter**

The `peer-filter` command places the switch in the peer-filter configuration mode and allows to create a peer-filter group. The peer-filter group parameters are defined using the `match as-range` command.

The `no peer-filter` or `default peer-filter` command removes the `peer-filter group` from the running configuration.

**Command Mode**
- Router-BGP Configuration
- Peer-Filter Configuration

**Command Syntax**

```
peer-filter filter_name
```

**Parameters**
- `filter_name`  name of the peer filter.

**Example**
- This command places the switch in the peer-filter configuration mode and creates a peer-filter "group1".

```
switch(config-router-bgp)#peer-filter group1
switch(config-peer-filter-group1)#
```
**rd (Router-BGP VRF and VNI Configuration Modes)**

The *rd* command adds a route distinguisher (RD) to VRF and VNI configuration modes. RDs internally identify routes belonging to a VRF or VNI to distinguish overlapping or duplicate IP address ranges. This allows the creation of distinct routes to the same IP address for different VPNs. The RD is a 64-bit number made up of an AS number or IPv4 address followed by a user-selected ID number.

If the switch is not running EVPN, an RD is not required for a VRF or VNI to function. Use `no` or `default` command forms to remove an RD from a VRF or VNI.

**Note** Legacy RDs that were assigned in VRF Configuration Mode appear in `show vrf` outputs if an RD has not been configured using this command, but they no longer have an effect on the system. RDs assigned in the VNI Configuration Mode are displayed in the output of `show bgp evpn` command.

**Command Modes**
- Router-BGP VRF Configuration
- Router-BGP VNI Configuration

**Command Syntax**

```
rd admin_ID:local_assignment
no rd
default rd
```

**Parameters**

- *admin_ID* An AS number or globally assigned IPv4 address identifying the entity assigning the RD. This should be an IANA-assigned identifying number.
- *local_assignment* A locally assigned number distinguishing the VRF. Values range from 0-65535 if the *admin_ID* is an IPv4 address, or from 0-4,294,967,295 if the *admin_ID* is an AS number. If the *admin_ID* is an AS number, the *local_assignment* can also be entered in the form of an IPv4 address.

**Examples**

- These commands identify the administrator of the VRF called “purple” as AS 530 and assign 12 as its local number.
  
  ```
  switch(config)#router bgp 50
  switch(config-router-bgp)#vrf purple
  switch(config-router-bgp-vrf-purple)#rd 530:12
  ```

- These commands identify the administrator of the MAC-VRF called “bundle1” as AS 530 and assign 12 as its local number.
  
  ```
  cvx(config)#router bgp 100
  cvx(config-router-bgp)#vni-aware-bundle bundle1
  cvx(config-macvrf-bundle1)#rd 530:12
  ```
redistribute (BGP)

The `redistribute` command enables the redistribution of specified routes to the BGP domain.

The `no redistribute` and `default redistribute` commands disable route redistribution from the specified domain by removing the corresponding `redistribute` command from `running-config`.

**Important!** Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

**Command Mode**
Router-BGP Configuration

**Command Syntax**
```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

**Parameters**
- **ROUTE_TYPE** source from which routes are redistributed. Options include:
  - `connected` routes that are established when IP is enabled on an interface.
  - `match nssa-external` all OSPF NSSA external routes.
  - `match nssa-external 1` type 1 OSPF NSSA external routes.
  - `match nssa-external 2` type 2 OSPF NSSA external routes.
  - `ospf` routes from an OSPF domain.
  - `ospf match external` routes external to the AS, but imported from OSPF.
  - `ospf match internal` OSPF routes that are internal to the AS.
  - `ospf match nssa-external` all OSPF NSSA external routes.
  - `ospf match nssa-external 1` type 1 OSPF NSSA external routes.
  - `ospf match nssa-external 2` type 2 OSPF NSSA external routes.
  - `ospf3` routes from an OSPFv3 domain.
  - `ospf3 match external` routes external to the AS, but imported from OSPFv3.
  - `ospf3 match internal` OSPFv3 routes that are internal to the AS.
  - `rip` routes from a RIP domain.
  - `static` IP static routes.
  - `isis` IS-IS routes. Sub-options include.
    - `level-1` Redistribute IS-IS level-1 routes
    - `level-1-2` Redistribute IS-IS level-1 and level-2 routes
    - `level-2` Redistribute IS-IS level-2 routes
    - `route-map` Route map reference

**Important!** While redistributing IS-IS routes into BGP, Level-1 or Level-2 keyword can be used to selectively redistribute Level-1 routes or Level-2 routes into BGP. The Level-1 or Level-2 keyword is optional and it defaults to Level-2 when not configured.

- **ROUTE_MAP** route map that determines the routes that are redistributed. Options include:
  - `<no parameter>` all routes are redistributed.
• **route-map map_name** only routes in the specified route map are redistributed.

**Example**

• This command redistributes OSPF routes into the BGP domain.

  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#redistribute OSPF
  switch(config-router-bgp)#
  ```

• These commands redistributes ISIS routes into BGP domain in address-family mode.

  ```
  Switch(config)#router bgp 1
  Switch(config-router-bgp)#address-family ipv4
  Switch(config-router-bgp-af)#redistribute isis level-1 route-map isis-to-bgp-v4
  ```

• These commands redistributes ISIS routes into BGP domain in router-bgp mode.

  ```
  Switch(config)#router bgp 1
  Switch(config-router-bgp)#redistribute isis level-1 route-map isis-to-bgp
  ```
router-id (BGP)

The `router-id` command sets the local router BGP router ID. When no ID has been specified, the local router ID is set to the following:

- The loopback IP address when a single loopback interface is configured.
- The loopback with the highest IP address when multiple loopback interfaces are configured.
- The highest IP address on a physical interface when no loopback interfaces are configured.

**Important!** The router-id must be specified if the switch has no IPv4 addresses configured.

The `no router-id` and `default router-id` commands remove the `router-id` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
router-id id_num
no router-id [id_num]
default router-id [id_num]
```

**Parameters**

- `id_num` router ID number (32-bit dotted decimal notation).

**Example**

- This command configures the fixed router ID address of 10.10.4.11

  `switch(config)#router bgp 9
   switch(config-router-bgp)#router-id 10.10.4.11
   switch(config-router-bgp)#`
router bgp

The `router bgp` command places the switch in router-BGP configuration mode. If BGP was not previously instantiated, this command creates a BGP instance with the specified AS number. Router-BGP configuration mode is not a group change mode; `running-config` is changed immediately after commands are executed. The `exit` command does not affect the configuration.

When a BGP instance exists, the command must include the AS number of the existing BGP instance. Running this command with a different AS number generates an error message.

The `no router bgp` and `default router bgp` commands delete the BGP instance.

Refer to Router-BGP Configuration Mode (Includes Address-Family Mode) (page 2226) for a list of commands available in router-BGP configuration mode.

The `exit` command returns the switch to global configuration mode.

**Command Mode**

Global Configuration

**Command Syntax**

```
router bgp as_id
no router bgp
default router bgp
```

**Parameters**

- `as_id` Autonomous system (AS) number. Values range from 1 to 4294967295.

**Examples**

- This command creates a BGP instance with AS number 64500.
  
  ```
  switch(config)#router bgp 64500
  switch(config-router-bgp)#
  ```

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.
  
  ```
  switch(config)#router bgp 64501
  % BGP is already running with AS number 64500
  switch(config)#
  ```

- This command exits BGP configuration mode.
  
  ```
  switch(config-router-bgp)#exit
  switch(config)#
  ```

- This command deletes the BGP instance.
  
  ```
  switch(config)#no router bgp
  switch(config)#
  ```
show bgp convergence

The `show bgp convergence` command displays information about the Border Gateway Protocol (BGP) convergence state and other statistics about the BGP instance in the specified VRF or in all VRFs.

**Command Mode**

EXEC

**Command Syntax**

```
show bgp convergence [VRF_INSTANCE]
```

**Parameters**

- **VRFINSTANCE** specifies VRF instances.
  - `<no parameter>` displays BGP information for the context-active VRF.
  - `vrf vrf_name` displays BGP information for the specified VRF.
  - `vrf all` displays BGP information for all VRFs.
  - `vrf default` displays BGP information for the default VRF.

**Examples**

- This command displays the output when no peers have joined before convergence.

  ```
  switch(config-router-bgp)#show bgp convergence
  BGP Convergence information for VRF: default
  Configured convergence timeout: 00:02:30
  Configured convergence slow peer timeout: 00:00:55
  Convergence based update synchronization is enabled
  Last Bgp convergence event : None
  Bgp convergence state : Not Initiated (Waiting for the first peer to join)
  Convergence timer is not running
  Convergence timeout in use: 00:02:30
  Convergence slow peer timeout in use: 00:00:55
  First peer is not up yet
  All the expected peers are up: no
  All IGP protocols have converged: yes
  Outstanding EORs: 0, Outstanding Keepalives: 0
  Pending Peers: 2
  Total Peers: 2
  Established Peers: 0
  Disabled Peers: 0
  Peers that have not converged yet:
  IPv4 peers:
  201.1.1.1 (Session : Connect)
  IPv6 peers:
  None
  ```
- This command displays the output when the first peer has joined before convergence.

  switch#show bgp convergence
  BGP Convergence information for VRF: default
  Configured convergence timeout: 00:02:30
  Configured convergence slow peer timeout: 00:00:55
  Convergence based update synchronization is enabled
  Last Bgp convergence event 00:00:40 ago
  Bgp convergence state : Pending (Waiting for EORs/Keepalives from peer(s) and IGP convergence)
    Convergence timer running, will expire in 00:01:50
    Convergence timeout in use: 00:02:30
    Convergence slow peer timeout in use: 00:00:55
    First peer came up 00:00:13 ago
    All the expected peers are up: no
    All IGP protocols have converged: yes
    Outstanding EORs: 0, Outstanding Keepalives: 0
    Pending Peers: 1
    Total Peers: 2
    Established Peers: 1
    Disabled Peers: 0
    Peers that have not converged yet:
    IPv4 peers:
      201.1.1.1 (Session : Active)
    IPv6 peers:
      None

- This command displays the output when the convergence timeout value is reached.

  switch(config-router-bgp)#show bgp convergence
  BGP Convergence information for VRF: default
  Configured convergence timeout: 00:02:30
  Configured convergence slow peer timeout: 00:00:55
  Convergence based update synchronization is enabled
  Last Bgp convergence event 00:02:44 ago
  Bgp convergence state : Timeout reached
    Time taken to converge 00:02:30
    Pending Peers: 1
    Total Peers: 2
    Established Peers: 1
    Disabled Peers: 0
    Peers that did not converge before local bgp convergence:
    IPv4 peers:
      201.1.1.1 (Session : Active)
      202.1.1.1 (Session : Established)
    IPv6 peers:
      None
This command displays the output during the converged state.

```
switch(config-router-bgp)#show bgp convergence
BGP Convergence information for VRF: default
Configured convergence timeout: 00:05:00
Configured convergence slow peer timeout: 00:01:30
Convergence based update synchronization is enabled
Last Bgp convergence event 00:00:05 ago
Bgp convergence state : Converged
  Time taken to converge 00:00:02
  First peer came up 00:00:05 ago
Pending Peers: 0
Total Peers: 3
Established Peers: 3
Disabled Peers: 0
Peers that did not converge before local bgp convergence:
  IPv4 peers:
    None
  IPv6 peers:
    None
```
show bgp instance

The show bgp instance command displays summary Border Gateway Protocol (BGP) information about the BGP instance in the specified VRF or in all VRFs.

Command Mode

EXEC

Command Syntax

show bgp instance [VRF_INSTANCE]

Parameters

- **VRF_INSTANCE** specifies VRF instances.
  - <no parameter> displays BGP information for the context-active VRF.
  - `vrf vrf_name` displays BGP information for the specified VRF.
  - `vrf all` displays BGP information for all VRFs.
  - `vrf default` displays BGP information for the default VRF.

Examples

- This command displays information about the BGP instance in the context-active VRF.

  switch>show bgp instance
  BGP instance information for VRF purple
  BGP Local AS: 64497, Router ID: 1.2.3.5
  Total peers: 5
  Configured peers: 3
  UnConfigured peers: 2
  Disabled peers: 0
  Established peers: 3
  Graceful restart helper mode enabled
  End of rib timer timeout: 00:05:00
  BGP Convergence timer is inactive
  BGP Convergence information:
    BGP has converged: no
    Outstanding EORs: 0, Outstanding Keepalives: 0
    Convergence timeout: 00:10:00
  switch>

- This command displays information about the BGP instance in the default VRF.

  switch>show bgp instance vrf default
  BGP instance information for VRF default
  BGP Local AS: 64503, Router ID: 1.2.3.5
  Total peers: 1
  Configured peers: 1
  UnConfigured peers: 0
  Disabled peers: 0
  Established peers: 0
  Graceful restart helper mode enabled
  End of rib timer timeout: 00:05:00
  BGP Convergence timer is inactive
  BGP Convergence information:
    BGP has converged: no
    Outstanding EORs: 0, Outstanding Keepalives: 0
    Convergence timeout: 00:10:00
  switch>
show ip as-path access-list

The show ip as-path access-list command displays BGP filters on the switch. Specifying an access list displays the statements from that access list. Entering the command without parameters displays the statements from all access lists on the switch.

Command Mode

EXEC

Command Syntax

show ip as-path access-list [list_name]

Parameters

- **list_name**: the name of an AS path access list.

Example

- This command displays the contents of the AS path access list named “list1.”

  switch>show ip as-path access-list list1
  ip as-path access-list list1 deny _3$
  ip as-path access-list list1 permit .*
  switch>
show ip bgp

The `show ip bgp` command displays Border Gateway Protocol (BGP) IPv4 routing table entries. The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries for the specified IPv4 addresses.

**Command Mode**

EXEC

**Command Syntax**

```
show ip bgp [FILTER] [VRF_INSTANCE]
```

**Parameters**

- `FILTER` routing table entries that the command displays. Options include:
  - `<no parameter>` displays all routing table entries in tabular format.
  - `detail` displays all routing table entries in data block format.
  - `ipv4_addr` displays IPv4 host address in data block format.
  - `ipv4_prefix` displays the route information of specified IPv4 prefix in data block format. Options include:
    - `detail` displays the detailed route information of specified IPv4 prefix in data block format.
    - `longer-prefixes` displays the route information of IPv4 prefix in tabular block format.
    - `longer-prefixes detail` displays the detailed route information of specified IPv4 prefix in data block format.
  - `community-list cmnty_list_name` displays BGP routes filtered by the specified community list.
  - `installed` displays the information of installed BGP routes.
  - `labeled-unicast` displays the information of labeled-unicast BGP routes only.
  - `not-installed` displays the information of BGP routes that are not installed.
- `VRF_INSTANCE` specifies VRF instances.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Guidelines**

You must provide the IPv4 prefix in CIDR notation.
Examples

- This command displays the BGP routing table.

```
switch#show ip bgp
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 10.1.0.0/24</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.1.0.0/24</td>
<td>10.1.0.100</td>
<td>42</td>
<td>100</td>
<td>0</td>
<td>64496 ?</td>
</tr>
<tr>
<td>* &gt; 10.2.0.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.3.0.0/24</td>
<td>10.3.0.103</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.100.0.0/24</td>
<td>10.1.0.100</td>
<td>200</td>
<td>100</td>
<td>0</td>
<td>64496 64497 65536 i</td>
</tr>
<tr>
<td>* &gt; 10.100.0.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.100.2.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.0.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.101.0.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.101.0.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.101.2.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.103.0.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.103.1.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.103.2.0/24</td>
<td>10.3.0.103</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
</tbody>
</table>
```

- This command displays the BGP routing table entry for the 10.100.1.0/24 network.

```
switch#show ip bgp 10.100.1.0/24
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 10.100.1.0/24
Paths: 1 available
64496 64497 65536
10.1.0.100 from 10.1.0.100 (10.0.0.100)
Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received 01:57:33 ago, valid, external, best
Community: 655:23590 64496:1000
Rx SAFI: Unicast
```

- This command displays the BGP routing table entry for the 10.105.1.0/24 network, including the reason why one route was discarded by the best-path algorithm. The reason for discarding a route is preceded by the label Not best:

```
switch#show ip bgp 10.105.1.0/24 detail
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 10.105.1.0/24
Paths: 2 available
64496
10.1.0.100 from 10.1.0.100 (10.0.0.100)
Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received 01:57:33 ago, valid, external, best
Community: 655:23590 64496:1000
Rx SAFI: Unicast
10.2.0.101 from 10.2.0.101 (12.0.0.101)
Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received 00:00:58 ago, valid, external, best
Community: 655:23590 64496:1000
Rx SAFI: Unicast
Not best: Origin
Advertised to 2 peers:
peer-group EXTERNAL: 10.3.0.103
peer-group INTERNAL: 10.3.0.103
switch#```
show ip bgp community

The **show ip bgp community** command displays Border Gateway Protocol (BGP) routing table entries, filtered by community.

**Command Mode**

EXEC

**Command Syntax**

```
show ip bgp community COMM_1 [COMM_2... COMM_n] [MATCH_TYPE] [DATA_OPTION] [VRF_INSTANCE]
```

**Parameters**

- **COMM_x** community number or name, as specified in the route map that sets the community list number.
- **GSHUT** well-known graceful shutdown community.
- **aa:nn** AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
- **comm_num** community number. Values range from 1 to 4294967040.
- **internet** advertises route to Internet community.
- **local-as** advertises route only to local peers.
- **no-advertise** does not advertise the route to any peer.
- **no-export** advertises route only within BGP AS boundary.
- **MATCH_TYPE** routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
  - **regex** display routes matching the regular expression of communities.
- **DATA_OPTION** type of information the command displays. Values include:
  - <no parameter> displays table of the routing entry line items.
  - **detail** displays data block for each routing table entry.
- **VRF_INSTANCE** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf vrf_name** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

**Guidelines**

The interpretation of regular expressions is always based on string mode but not on the ACL configuration.
Example

- This command displays the BGP routing table entries with the community 64496:1000.

  switch#show ip bgp community 64496:1000 detail
  BGP routing table information for VRF default
  Router identifier 10.0.0.102, local AS number 64500
  BGP routing table entry for 10.100.1.0/24
  Paths: 1 available
    64496 64497 65536
    10.1.0.100 from 10.1.0.100 (10.0.0.100)
    Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received 00:03:16 ago, valid, external, best
    Community: 655123590 64496:1000
    Rx SAFI: Unicast

  switch#
show ip bgp neighbors

The **show ip bgp neighbors** command displays Border Gateway Protocol (BGP) and TCP session data for a specified IPv4 BGP neighbor, or for all IPv4 BGP neighbors if an address is not included.

**Command Mode**

EXEC

**Command Syntax**

```
show ip bgp neighbors [NEIGHBOR_ADDR][VRF_INSTANCE]
```

**Parameters**

- **NEIGHBOR_ADDR** location of the neighbors. Options include:
  - **<no parameter>** command displays information for all IPv4 BGP neighbors.
  - **ipv4_addr** command displays information for specified neighbor.
- **VRF_INSTANCE** specifies VRF instances.
  - **<no parameter>** displays routing table for context-active VRF.
  - **vrf vrf_name** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

**Related Command**

- **show ip bgp neighbors (route type)**
- **show ip bgp neighbors (route-type) community**
Example

- This command displays information of the neighbor at 10.1.0.100.

```
switch(config)#show ip bgp neighbors 10.1.0.100
BGP neighbor is 10.1.0.100, remote AS 64496, external link
  BGP version 4, remote router ID 10.0.0.100, VRF default
  Inherits configuration from and member of peer-group EXTERNAL
  Negotiated BGP version 4
  Member of update group 3
  Last read 00:00:17, last write 00:00:18
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 00:05:17
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was RecvKeepAlive

Neighbor Capabilities:
  Multiprotocol IPv4 Unicast: advertised and received and negotiated
  Four Octet ASN: advertised and received and negotiated
  Route Refresh: advertised and received and negotiated
  Send End-of-RIB messages: advertised and received and negotiated
  Additional-paths recv capability:
    IPv4 Unicast: advertised
  Additional-paths send capability:
    IPv4 Unicast: received

Restart timer is inactive
End of rib timer is inactive

Message Statistics:
  InQ depth is 0
  OutQ depth is 0

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Notifications</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Updates</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Keepalives</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Route-Refresh</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total messages</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Prefix Statistics:
  IPv4 Unicast: 9 4
  IPv6 Unicast: 0 0
  IPv4 SR-TE: 0 0
  IPv6 SR-TE: 0 0

Inbound updates dropped by reason:
  AS path loop detection: 0
  Enforced First AS: 0
  Originator ID matches local router ID: 0
  Next-hop matches local IP address: 0
  Unexpected IPv6 next-hop for IPv4 routes: 0
  Next-hop invalid for single hop eBGP: 0

Inbound updates with attribute errors:
  Resulting in removal of all paths in update (treat-as-withdraw): 0
  Resulting in AFI/SAFI disable: 0
  Resulting in attribute ignore: 0

Inbound paths dropped by reason:
  IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
  IPv6 labeled-unicast NLRIs dropped due to excessive labels: 0

Outbound paths dropped by reason:
  IPv4 local address not available: 0
  IPv6 local address not available: 0

Local AS is 64500, local router ID 10.0.0.102
TTL is 255, BGP neighbor may be upto 1 hops away
Local TCP address is 10.1.0.102, local port is 179
Remote TCP address is 10.1.0.100, remote port is 33171
Auto-Local-Addr is disabled

TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 1448
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
    Window Scale (wscale): 9,9
Retransmission Timeout (rto): 204.0ms
Round-trip Time (rtt/rtvar): 3.0ms/5.4ms
Delayed Ack Timeout (ato): 40.0ms
Congestion Window (cwnd): 10
TCP Throughput: 39.20 Mbps
Advertised Recv Window (rcv_space): 28960

switch(config)#
show ip bgp neighbors (route type)

The `show ip bgp neighbors (route type)` command displays information for next hop routes to a specified IPv4 neighbor. The `show ip bgp neighbors (route-type) community` command displays the same information for routes filtered by communities.

The output format depends on the selected `FILTER` parameter:
- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Commands that do not include a route type revert to the `show ip bgp neighbors` command.

Command Mode

EXEC

Command Syntax

```
show ip bgp neighbors neighbor_addr HOPDIRECT [FILTER] [VRF_INSTANCE]
show ip bgp neighbors neighbor_addr [ROUTE_TYPE] HOPDIRECT
show ip bgp neighbors neighbor_addr [ROUTE_TYPE] HOPDIRECT detail
```

Related Command
- `show ip bgp neighbors`
- `show ip bgp neighbors (route-type) community`

Parameters
- `neighbor_addr` location of the neighbor.
- `ROUTE_TYPE` filters route on route type. Options include:
  - `ipv4 unicast` displays IPv4 unicast routes.
  - `ipv6 unicast` displays IPv6 unicast routes.
- `HOPDIRECT` filters route on the basis of direction from neighbor. Options include:
  - `advertised-routes` displays routes advertised to the specified neighbor.
  - `received-routes` displays routes received from the specified neighbor (accepted and rejected).
  - `routes` displays routes received and accepted from specified neighbor.
- `FILTER` routing table entries that the command displays. Values include:
  - `<no parameter>` displays all routing table entries. Tabular format.
  - `detail` displays all routing table entries. Data block format.
  - `ipv4_addr` displays IPv4 host address in data block format.
  - `ipv4_prefix` displays the route information of specified IPv4 prefix in data block format. Option includes:
    - `longer-prefixes` displays the route information of IPv4 prefix in tabular block format.
- `VRF_INSTANCE` specifies VRF instances.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.
Example

- This command displays information for routes advertised to the neighbor at 10.3.0.103.

```bash
switch#show ip bgp neighbors 10.3.0.103 advertised-routes
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, l - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 10.1.0.0/24</td>
<td>10.3.0.102</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.2.0.0/24</td>
<td>10.3.0.102</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>i</td>
</tr>
<tr>
<td>* &gt; 10.3.0.0/24</td>
<td>10.3.0.102</td>
<td>200</td>
<td>100</td>
<td>-</td>
<td>65536 i</td>
</tr>
<tr>
<td>* &gt; 10.100.0.0/24</td>
<td>10.1.0.100</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>64496 64497 65536 i</td>
</tr>
<tr>
<td>* &gt; 10.100.1.0/24</td>
<td>10.1.0.100</td>
<td>42</td>
<td>100</td>
<td>-</td>
<td>64496 ?</td>
</tr>
<tr>
<td>* &gt; 10.101.0.0/24</td>
<td>10.2.0.101</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.1.0/24</td>
<td>10.2.0.101</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.2.0/24</td>
<td>10.2.0.101</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>64510 i</td>
</tr>
</tbody>
</table>

switch#
```
show ip bgp neighbors (route-type) community

The `show ip bgp neighbors (route type) community` command displays information for next hop routes to a specified neighbor. Routes are filtered by community.

The `show ip bgp neighbors (route type)` command displays the same information for routes filtered by IP addresses and subnets.

**Command Mode**

EXEC

**Command Syntax**

```
show ip bgp neighbors addr RTE community CM_1 [CM_2...CM_n] [MATCH] [INFO] [VRF_INST]
```

**Related Command**

- `show ip bgp neighbors`
- `show ip bgp neighbors (route type)`

**Parameters**

- `addr` location of the neighbor.
- `RTE` type of route that the command displays. Options include:
  - `advertised-routes` displays routes advertised to the specified neighbor.
  - `received-routes` displays routes received from the specified neighbor (accepted and rejected).
  - `routes` displays routes received and accepted from specified neighbor.
- `CM_x` community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:
  - `GSHUT` well-known graceful shutdown community.
  - `aa:nn` AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - `comm_num` community number. Values range from 1 to 4294967040.
  - `internet` advertises route to Internet community.
  - `local-as` advertises route only to local peers.
  - `no-advertise` does not advertise route to any peer.
  - `no-export` advertises route only within BGP AS boundary.
- `MATCH` Routes are filtered based on their communities.
  - `<no parameter>` routes must match at least one community in the list
  - `exact` route must match all communities and include no other communities.
- `INFO` Type of information the command displays. Values include:
  - `<no parameter>` Displays table of routing entry line items.
  - `detail` Displays data block for each routing table entry.
- `VRF_INST` specifies VRF instances.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
• **vrf default** displays routing table for default VRF.

**Example**

- This command lists the routes advertised to the neighbor at 10.3.0.103 with community 655:23590.

```
switch#show ip bgp neighbors 10.3.0.103 advertised-routes community 655:23590
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 10.100.1.0/24</td>
<td>10.1.0.100</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>64496 64497 65536 i</td>
</tr>
</tbody>
</table>
```

switch#
**show ip bgp neighbors regexp**

The `show ip bgp neighbors regexp` command displays information for next hop routes to a specified IPv4 neighbor that match the AS path attributes specified in the given regular expression.

**Command Mode**

EXEC

**Command Syntax**

```
show ip bgp neighbors addr RTE regexp as_paths [VRF_INST]
```

**Related Command**

- `show ip bgp neighbors`
- `show ip bgp neighbors (route type)`

**Parameters**

- `addr` location of the neighbor.
- `RTE` type of route that the command displays. Options include:
  - `advertised-routes` displays routes advertised to the specified neighbor.
  - `received-routes` displays routes received from the specified neighbor (accepted and rejected).
  - `routes` displays routes received and accepted from specified neighbor.
- `as_paths` list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- `VRF_INST` specifies VRF instances.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Example**

- This command lists the routes advertised to the neighbor at 10.3.0.103 where the AS path is 64496.

```bash
switch#show ip bgp neighbors 10.3.0.103 advertised-routes regex ^64496$
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L = labeled-unicast
% - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 10.100.0.0/24</td>
<td>10.1.0.100</td>
<td>200</td>
<td>100</td>
<td>-</td>
<td>64496 i</td>
</tr>
<tr>
<td>* &gt; 10.100.2.0/24</td>
<td>10.1.0.100</td>
<td>42</td>
<td>100</td>
<td>-</td>
<td>64496 ?</td>
</tr>
</tbody>
</table>
```

**switch#**
**show ip bgp paths**

The `show ip bgp paths` command displays all BGP AS paths in the database.

**Command Mode**

EXEC

**Command Syntax**

`show ip bgp paths [VRF_INSTANCE]`

**Parameters**

- **VRF_INSTANCE** specifies VRF instances.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Display Values**

- **Refcount**: Number of routes using a listed path.
- **Metric**: The path’s Multi Exit Discriminator (MED).
- **Path**: The route’s AS path and its origin code.

The MED (the path’s external metric) provides information to external neighbors about the preferred path into an AS that has multiple entry points. Lower MED values are preferred.

**Example**

- This command displays all BGP AS paths in the switch’s database.

```
switch#show ip bgp paths
Refcount Metric     Path
 6        0          64510 64505 64506 64507 i (HashID 9)
 6        0          64510 ? (HashID 8)
12       0          65530 65531 65532 e (HashID 5)
12       0          i (HashID 6)
 6        0          64100 64200 i (HashID 4)
 28       0          i (HashID 1)
 7        0          ? (HashID 2)
40       0          64510 i (HashID 10)
19       0          64510 i (HashID 7)
 2        0          i (HashID 3)
switch#
```
show ip bgp peer-group

The show ip bgp peer-group command displays the BGP version, address family and group members for all BGP peer groups defined on the switch.

Command Mode

EXEC

Command Syntax

show ip bgp peer-group [GROUP] [VRF_INSTANCE]

Parameters

- **GROUP** peer group for which command displays information. Options include:
  - <no parameter> command displays information for all peer groups.
  - `group_name` name of peer group for which command displays information.
- **VRF_INSTANCE** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

Example

- This command displays BGP peer group information for all peer groups on the switch.

```
switch#show ip bgp peer-group
BGP peer-group is EXTERNAL
BGP version 4
Static peer-group members:
  VRF default:
    10.1.0.100, state: Connect
    Negotiated MP Capabilities:
      IPv4 Unicast: No
      IPv6 Unicast: No
      IPv4 SR-TE: No
      IPv6 SR-TE: No
    10.2.0.101, state: Connect
    Negotiated MP Capabilities:
      IPv4 Unicast: No
      IPv6 Unicast: No
      IPv4 SR-TE: No
      IPv6 SR-TE: No
BGP peer-group is INTERNAL
BGP version 4
Listen-range subnets:
  VRF default:
    10.3.0.0/24, remote AS 64500
Dynamic peer-group members:
  VRF default:
switch#
```
show ip bgp regexp

The `show ip bgp regexp` command displays Border Gateway Protocol (BGP) IPv4 routing table entries that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

```
show ip bgp regexp as_paths [VRF_INSTANCE]
```

Parameters

- `as_paths` list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- `VRF_INSTANCE` specifies the VRF instance of the BGP routing table to be displayed.
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

Example

- This command displays information about the BGP IPv4 routes in the context-active VRF where the AS path is 64510.

```
switch#show ip bgp regex ^64510$
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L = labeled-unicast
% - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2.0.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.0.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.1.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
<tr>
<td>* &gt; 10.101.2.0/24</td>
<td>10.2.0.101</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64510 i</td>
</tr>
</tbody>
</table>

switch#
```
show ip bgp summary

The `show ip bgp summary` command displays the summary of all IPv4 and IPv6 BGP neighbors based on exchanged Address Family Identifiers (AFI) and Subsequent Address Family Identifiers (SAFI) negotiations where AFI is ‘IP’ and SAFI is ‘unicast’ information.

**Command Mode**

EXEC

**Command Syntax**

`show ip bgp summary [VRF_INSTANCE]`

**Parameters**

- `VRF_INSTANCE` specifies VRF instances.

- `<no parameter>` displays routing table for context-active VRF.

- `vrf vrf_name` displays routing table for the specified VRF.

- `vrf all` displays routing table for all VRFs.

- `vrf default` displays routing table for default VRF.

**Display Values**

**Header Row**

- BGP router identifier: The router identifier – loopback address or highest IP address.

- Local AS Number: AS number assigned to switch

**Neighbor Table Columns**

- (First) Neighbor: Neighbor’s IP address.

- (Second) V: BGP version number.

- (Third) AS: Neighbor’s AS number.

- (Fourth) MsgRcvd: Messages received from the neighbor.

- (Fifth) MsgSent: Messages sent to neighbor.

- (Sixth) InQ: Messages queued from neighbor.

- (Seventh) OutQ: Messages queued to send neighbor.

- (Eighth) Up/Down: Period the BGP session has been Established, or its current status.

- (Ninth) State: State of the BGP session and the number of routes received from a neighbor.

  After the maximum number of routes are received, the ninth field displays `PfxRcd`, and the connection becomes Idle. Maximum number of routes is set using the `maximum paths (BGP)` command.

**Related Commands**

- `show ipv6 bgp summary`

**Example**

- This command displays the status of the switch’s BGP connections.

```
switch#show ip bgp summary
BGP summary information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Neighbor Status Codes: m - Under maintenance
  Neighbor    V  AS     MsgRcvd  MsgSent  InQ OutQ  Up/Down State  PfxRcd  PfxAcc
10.1.0.100    4 64496   1075     1083    0    0 00:04:04 Connect
10.2.0.101    4 64510   1079     1088    0    0 00:04:14 Connect
switch#
```
show ip community-list

The show ip community-list command displays the BGP community lists configured on the switch.

Command Mode

EXEC

Command Syntax

show ip community-list [COMMUNITY_LIST]

Parameters

- **COMMUNITY_LIST**  community list for which command displays information.
- <no parameter>  command displays information for all community lists.
- **listname**  name of the community list (text string).

Example

- This command displays the BGP paths in the switch’s database.

  switch>show ip community-list hs-comm-list
  ip community-list hs-comm-list permit 0:10
  switch>
show ip extcommunity-list

The show ip extcommunity-list command displays the BGP extended community lists configured on the switch.

Command Mode
EXEC

Command Syntax
show ip extcommunity-list [COMMUNITY_LIST]

Parameters
- COMMUNITY_LIST extended community list for which command displays information.
- <no parameter> command displays information for all extended community lists.
- listname name of the extended community list (text string).

Example
- This command displays the extended extcommunity lists on the switch.
  switch>show ip extcommunity-list
  ip extcommunity-list hs-extcomm-list permit rt 3050:20
  ip extcommunity-list hs-extcomm-list permit soo 172.17.52.2:30
  ip extcommunity-list hs-extcomm-list permit rt 3050:70000
  switch>
show ipv6 bgp

The `show ipv6 bgp` command displays IPv6 Border Gateway Protocol (BGP) routing table entries. The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries for specified IPv6 addresses.

**Command Mode**

EXEC

**Command Syntax**

```
show ipv6 bgp [FILTER][VRF_INSTANCE]
```

**Parameters**

- **FILTER** routing table entries that the command displays. Options include:
  - `<no parameter>` displays all routing table entries in tabular format.
  - `detail` displays all routing table entries in data block format.
  - `ipv6_addr` displays IPv6 host address in data block format.
  - `ipv6_prefix` displays the route information of specified IPv6 prefix address in data block format. Options include:
    - `detail` displays the detailed route information of specified IPv6 prefix address in data block format.
    - `longer-prefixes` displays the route information of IPv6 prefix in tabular block format.
    - `longer-prefixes detail` displays the detailed route information of specified IPv6 prefix in data block format.
  - `community-list cmnty_list_name` displays BGP routes filtered by the specified community list.
  - `installed` displays the information of installed BGP routes.
  - `labeled-unicast` displays the information of labeled-unicast BGP routes only.
  - `not-installed` displays the information of BGP routes that are not installed.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Guidelines**

You must provide the IPv6 prefix in CIDR notation.

**Related Command**

- `show ip bgp`
Examples

- This command displays the route information of 2001:10:1:0::102/64 in data block format.

```
switch#show ipv6 bgp 2001:10:1:0::102/64
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 2001:10:1::/64
Paths: 2 available
 Local
   - from - (10.0.0.102)
      Origin IGP, metric 1, localpref 0, IGP metric -, weight -, received 00:16:27 ago, valid, local, best,
     redistributed (Connected)
     Rx SAFI: Unicast
64496
  2001:10:1::100 from 2001:10:1::100 (10.0.0.100)
     Origin INCOMPLETE, metric 42, localpref 100, IGP metric 1, weight 0, received 00:10:09 ago, valid,
    external
     Rx SAFI: Unicast
switch#
```
**show ipv6 bgp match community**

The *show ipv6 bgp match community* command displays IPv6 Border Gateway Protocol (BGP) routing table entries, filtered by community.

**Command Mode**

EXEC

**Command Syntax**

```
show ipv6 bgp match community [COMM_1 ... COMM_n][MATCH_TYPE][INFO][VRF_INSTANCE]
```

**Parameters**

- **COMM_x** community number or name, as specified in the route map that sets the community list number. Options include:
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - *internet* advertises route to Internet community.
  - *local-as* advertises route only to local peers.
  - *no-advertise* does not advertise route to any peer.
  - *no-export* advertises route only within BGP AS boundary.

- **MATCH_TYPE** routes are filtered based on their communities. Options include:
  - <no parameter> routes must match at least one community in the list.
  - *exact* route must match all communities and include no other communities.

- **INFO** type of information the command displays. Values include:
  - <no parameter> displays table of the routing entry line items.
  - *detail* displays data block for each routing table entry.

- **VRF_INSTANCE** specifies VRF instances. Options include:
  - <no parameter> displays routing table for context-active VRF.
  - *vrf vrf_name* displays routing table for the specified VRF.
  - *vrf all* displays routing table for all VRFs.
  - *vrf default* displays routing table for default VRF.

**Example**

- This command displays data block for each routing table entry with community 655:23590.

```bash
switch(config)#show ipv6 bgp match community 655:23590 detail
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
BGP routing table entry for 2001:10:100:1::/64
Paths: 1 available
64496 64497 65536
2001:10:1::100 from 2001:10:1::100 (10.0.0.100)
Origin IGP, metric 0, localpref 100, IGP metric 1, weight 0, received 01:09:29 ago, valid, external, best
Community: 655:23590 64496:1000
Rx SAFI: Unicast
switch(config)#
```
show ipv6 bgp peers

The `show ipv6 bgp peers` command displays IPv6 Border Gateway Protocol (BGP) and TCP session data for a specified neighbor. Command displays data for all neighbors if an address is not included.

**Command Mode**

EXEC

**Command Syntax**

```
show ipv6 bgp peers [NEIGHBOR_ADDR] [VRF_INSTANCE]
```

**Parameters**

- **NEIGHBOR_ADDR** location of the neighbors. Options include:
  - `<no parameter>` command displays information for all neighbors.
  - `ipv6_addr` command displays information for specified neighbor.
- **VRF_INSTANCE** specifies VRF instances. Options include:
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Related Commands**

- `show ip bgp peer-group`
Example

- This command displays information of the neighbor at 2001:10:1:0::100.

```
switch>show ipv6 bgp peers 2001:10:1:0::100
BGP neighbor is 2001:10:1::100, remote AS 64496, external link
BGP version 4, remote router ID 10.0.0.100, VRF default
Inherits configuration from and member of peer-group EXTERNAL
Negotiated BGP version 4
Member of update group 3
Last read 00:00:01, last write 00:00:01
Hold time is 180, keepalive interval is 60 seconds
Configured hold time is 180, keepalive interval is 60 seconds
Connect timer is inactive
Idle-restart timer is inactive
BGP state is Established, up for 00:12:01
Number of transitions to established: 1
Last state was OpenConfirm
Last event was RecvKeepAlive
Neighbor Capabilities:
  Multiprotocol IPv6 Unicast: advertised and received and negotiated
  Four Octet ASN: advertised and received and negotiated
  Route Refresh: advertised and received and negotiated
  Send End-of-RIB messages: advertised and received and negotiated
  Additional-paths recv capability:
    IPv6 Unicast: advertised
  Additional-paths send capability:
    IPv6 Unicast: received
Restart timer is inactive
End of rib timer is inactive
Message Statistics:
  InQ depth is 0
  OutQ depth is 0

<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
</table>
  Opens: | 1     | 1     |
  Notifications: | 0     | 0     |
  Updates: | 4     | 5     |
  Keepalives: | 14    | 14    |
  Route-Refresh: | 0     | 0     |
  Total messages: | 19    | 20    |
Prefix Statistics:

<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
</table>
  IPv4 Unicast: | 0     | 0     |
  IPv6 Unicast: | 6     | 4     |
  IPv4 SR-TE: | 0     | 0     |
  IPv6 SR-TE: | 0     | 0     |
Inbound updates dropped by reason:
  AS path loop detection: 0
  Enforced First AS: 0
  Originator ID matches local router ID: 0
  Nexthop matches local IP address: 0
  Unexpected IPv6 nexthop for IPv4 routes: 0
  Nexthop invalid for single hop eBGP: 0
Inbound updates with attribute errors:
  Resulting in removal of all paths in update (treat-as-withdraw): 0
  Resulting in AFI/SAFI disable: 0
  Resulting in attribute ignore: 0
Inbound paths dropped by reason:
  IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
  IPv6 labeled-unicast NLRIs dropped due to excessive labels: 0
```
Outbound paths dropped by reason:
IPv4 local address not available: 0
IPv6 local address not available: 0
Local AS is 64500, local router ID 10.0.0.102
TTL is 1
Local TCP address is 2001:10:1::102, local port is 45983
Remote TCP address is 2001:10:1::100, remote port is 179
Auto-Local-Addr is disabled
TCP Socket Information:
TCP state is ESTABLISHED
Recv-Q: 0/32768
Send-Q: 0/32768
Outgoing Maximum Segment Size (MSS): 1428
Total Number of TCP retransmissions: 0
Options:
Timestamps enabled: yes
Selective Acknowledgments enabled: yes
Window Scale enabled: yes
Explicit Congestion Notification (ECN) enabled: no
Socket Statistics:
Window Scale (wscale): 9,9
Retransmission Timeout (rto): 204.0ms
Round-trip Time (rtt/rtvar): 1.4ms/2.7ms
Delayed Ack Timeout (ato): 40.0ms
Congestion Window (cwnd): 10
TCP Throughput: 80.00 Mbps
Advertised Recv Window (rcv_space): 28800
switch>
show ipv6 bgp peers (route type)

The show ipv6 bgp peers (route type) command displays information about the routes either advertised to or received from a specified IPv6 BGP neighbor. The show ipv6 bgp peers (route type) community command displays the same information for routes filtered by communities. Commands that do not include a route type revert to the show ipv6 bgp peers command.

The output format depends on the selected FILTER parameter:
- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Output produced by the longer-prefixes option includes the specified route and all more specific routes.

Command Mode
- EXEC

Command Syntax
- show ipv6 bgp peers neighbor_addr HOPDIRECT [FILTER] [VRF_INSTANCE]
- show ipv6 bgp peers neighbor_addr [ROUTE_TYPE] HOPDIRECT
- show ipv6 bgp peers neighbor_addr [ROUTE_TYPE] HOPDIRECT detail

Parameters
- neighbor_addr: location of the neighbor.
- ROUTE_TYPE: filters route on route type. Options include:
  - ipv4 unicast: displays IPv4 unicast routes.
  - ipv6 unicast: displays IPv6 unicast routes.
- HOPDIRECT: filters route on the basis of direction from neighbor. Options include:
  - advertised-routes: displays routes advertised to the specified neighbor.
  - received-routes: displays routes received from the specified neighbor (accepted and rejected).
  - routes: displays routes received and accepted from specified neighbor.
- FILTER: routing table entries that the command displays. Values include:
  - <no parameter>: displays all routing table entries. Tabular format.
  - detail: displays all routing table entries. Data block format.
  - ipv6_addr: displays the IPv6 host address in data block format.
  - ipv6_prefix: displays the route information of specified IPv6 prefix in data block format. Option includes:
    - longer-prefixes: displays the route information of IPv4 prefix in tabular block format.
- VRF_INSTANCE: specifies VRF instances. Options include:
  - <no parameter>: displays routing table for context-active VRF.
  - vrf vrf_name: displays routing table for the specified VRF.
  - vrf all: displays routing table for all VRFs.
  - vrf default: displays routing table for default VRF.

Related Commands
- show ipv6 bgp peers (route type) community
Example

- This command displays information of all routes advertised to the neighbor at 2001:10:1:0::100.

switch# show ipv6 bgp peers 2001:10:1:0::100 advertised-routes
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 2001:10:1::/64</td>
<td>2001:10:1:1:102</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64500  i</td>
</tr>
<tr>
<td>* &gt; 2001:10:2::/64</td>
<td>2001:10:1:1:102</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64500  i</td>
</tr>
<tr>
<td>* &gt; 2001:10:3::/64</td>
<td>2001:10:1:1:102</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64500  i</td>
</tr>
</tbody>
</table>

switch#
**show ipv6 bgp peers (route type) community**

The `show ipv6 bgp peers (route type) community` command displays information about the routes either advertised to or received from a specified IPv6 BGP neighbor. The routes are filtered by community.

The `show ipv6 bgp peers (route type)` command displays the same information for routes filtered by IP addresses and prefixes.

**Command Mode**

EXEC

**Command Syntax**

```
show ipv6 bgp peers  adr  RTE  community CM_1 [CM_2..CM_n][MATCH][INFO][VRF_INST]
```

**Parameters**

- **adr** Neighbor location (IPv6 address).
- **RTE** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- **CM_x** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:
  - **GSHUT** well-known graceful shutdown community.
  - **aa:nn** AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - **comm_num** community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- **MATCH** Routes are filtered based on their communities. Options include:
  - `<no parameter>` routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- **INFO** Type of information the command displays. Values include:
  - `<no parameter>` Displays table of the routing entry line items.
  - **detail** Displays data block for each routing table entry.
- **VRF_INST** specifies VRF instances. Options include:
  - `<no parameter>` displays routing table for context-active VRF.
  - **vrf vrf_name** displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

**Related Commands**

- **show ipv6 bgp peers**
Example

- This command lists the routes advertised to the neighbor at 2001:10:1:0::102 with the community 64496:1000.

```
switch>show ipv6 bgp peers 2001:10:1:0::102 advertised-routes community 64496:1000
```

BGP routing table information for VRF default
Router identifier 10.0.0.100, local AS number 64496
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

```
<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt; 2001:10:100:1::/64</td>
<td>2001:10:1:100</td>
<td></td>
<td></td>
<td></td>
<td>64496 64497 65536 i</td>
</tr>
</tbody>
</table>
```

switch>
show ipv6 bgp peers regexp

The `show ipv6 bgp peers regexp` command displays information about routes (advertised or received) from a specified IPv6 neighbor that match the AS path attributes specified in the given regular expression.

**Command Mode**

EXEC

**Command Syntax**

```
show ipv6 bgp peers addr RTE regexp as_paths [VRF_INST]
```

**Parameters**

- `addr` Neighbor location (IPv6 address).
- `RTE` type of route that the command displays. Options include:
  - `advertised-routes` displays routes advertised to the specified neighbor.
  - `received-routes` displays routes received from the specified neighbor (accepted and rejected).
  - `routes` displays routes received and accepted from specified neighbor.
- `as_paths` list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- `VRF_INST` specifies VRF instances. Options include:
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Related Command**

- `show ip bgp regexp`
- `show ipv6 bgp peers`

**Examples**

- This command displays information of routes received from the neighbor at 2001:10:1::100 which include AS number 64496 in their AS paths.

```
switch>show ipv6 bgp peers 2001:10:1::100 received-routes regex 64496
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 2001:10:1::64</td>
<td>2001:10:1:1:100</td>
<td>42</td>
<td>-</td>
<td>-</td>
<td>64496 ?</td>
</tr>
<tr>
<td>* &gt; 2001:10:1:100:1::64</td>
<td>2001:10:1:1:100</td>
<td>200</td>
<td>-</td>
<td>-</td>
<td>64496 i</td>
</tr>
<tr>
<td>* &gt; 2001:10:1::64</td>
<td>2001:10:1:1:100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64496 64497 65536 i</td>
</tr>
<tr>
<td>* &gt; 2001:10:1::2:64</td>
<td>2001:10:1:1:100</td>
<td>42</td>
<td>-</td>
<td>-</td>
<td>64496 ?</td>
</tr>
</tbody>
</table>
```

switch>
**show ipv6 bgp regexp**

The **show ipv6 bgp regexp** command displays Border Gateway Protocol (BGP) IPv6 routing table entries that match the AS path attributes specified in the given regular expression.

**Command Mode**

EXEC

**Command Syntax**

`show ipv6 bgp regexp as_paths [VRF_INSTANCE]`

**Parameters**

- **as_paths** list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- **VRF_INSTANCE** specifies the VRF instance of the BGP routing table to be displayed. Options include:
  - <no parameter> displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

**Related Command**

- **show ip bgp regexp**

**Examples**

- This command displays information about the BGP IPv6 routes in the context-active VRF that pass through AS 64496.

```
switch>show ipv6 bgp regexp 64496
BGP routing table information for VRF default
Router identifier 10.0.0.2, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP
S - Stale, c - Contributing to ECMP, b - backup, L = labeled-unicast
% - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST -Cluster List, LL Nexthop - Link Local Nexthop

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPref</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 2001:10:1::/64</td>
<td>2001:10:1:100</td>
<td>42</td>
<td>100</td>
<td>0</td>
<td>64496 ?</td>
</tr>
<tr>
<td>* &gt; 2001:10:100::/64</td>
<td>2001:10:1:100</td>
<td>200</td>
<td>100</td>
<td>0</td>
<td>64496 i</td>
</tr>
<tr>
<td>* &gt; 2001:10:100:1::/64</td>
<td>2001:10:1:100</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64496 64497 65536 i</td>
</tr>
<tr>
<td>* &gt; 2001:10:100:2::/64</td>
<td>2001:10:1:100</td>
<td>42</td>
<td>100</td>
<td>0</td>
<td>64496 ?</td>
</tr>
</tbody>
</table>
```

switch>
show ipv6 bgp summary

The `show ipv6 bgp summary` command displays the summary of all IPv4 and IPv6 BGP neighbors based on exchanged Address Family Identifiers (AFI) and Subsequent Address Family Identifiers (SAFI) negotiations where AFI is ‘IPv6’ and SAFI is ‘Unicast’ information.

Command Mode
EXEC

Command Syntax
`show ipv6 bgp summary [VRF_INSTANCE]`

Parameters
- `VRF_INSTANCE` specifies VRF instances. Options include:
  - `<no parameter>` displays routing table for context-active VRF.
  - `vrf vrf_name` displays routing table for the specified VRF.
  - `vrf all` displays routing table for all VRFs.
  - `vrf default` displays routing table for default VRF.

Display Values

Header Row
- **BGP router identifier**: The router identifier - loopback address or highest IP address.
- **Local AS Number**: AS number assigned to switch.

Neighbor Table Columns
- **(First) Neighbor**: Neighbor's IP address.
- **(Second) V**: BGP version number.
- **(Third) AS**: Neighbor's AS number.
- **(Fourth) MsgRcvd**: Messages received from the neighbor.
- **(Fifth) MsgSent**: Messages sent to neighbor.
- **(Sixth) InQ**: Messages queued from neighbor.
- **(Seventh) OutQ**: Messages queued to send neighbor.
- **(Eighth) Up/Down**: Period the BGP session has been Established, or its current status.
- **(Ninth) State**: State of the BGP session and the number of routes received from a neighbor.
- **(Tenth) PfxRcd**: The count of prefixes received by BGP per neighbor.
- **(Eleventh) PfxAcc**: The count of prefixes added to the BGP RIB among all received prefixes.

Related Commands
- `show ip bgp summary`

Example
- This command displays the status of the switch’s BGP connections.

```
switch>show ipv6 bgp summary
BGP summary information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Neighbor Status Codes: m - Under maintenance
Neighbor        V  AS    MsgRcvd  MsgSent  InQ OutQ  Up/Down  State   PfxRcd  PfxAcc
2001:10:1::100  4  64496  37     36      0    0 00:29:33 Estab  4   4
2001:10:2::101  4  64510  35     38      0    0 00:29:37 Estab  4   4
switch>
```
show peer-filter

The show peer-filter command displays the definition of a peer filter.

Command Mode
   EXEC

Command Syntax
   show peer-filter  filter_name

Parameters
   •  filter_name  name of the peer filter group.

Example
   •  This command displays the peer-filter group information.

switch#show peer-filter group3
peer-filter group3
   10 match as-range 65003 result accept
   20 match as-range 65007 result accept
   30 match as-range 65009 result accept
shutdown (BGP)

The `shutdown` command disables BGP on the switch without modifying the BGP configuration. The `no shutdown` and `default shutdown` commands enable the BGP instance by removing the `shutdown` command from `running-config`.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
shutdown
nenoshutdown
defaultshutdown
```

**Examples**

- This command disables BGP on the switch.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#shutdown
  switch(config-router-bgp)#
  ```

- This command enables BGP on the switch.
  ```
  switch(config)#router bgp 9
  switch(config-router-bgp)#no shutdown
  switch(config-router-bgp)#
  ```
timers bgp

The `timers bgp` command configures the BGP keepalive and hold times. Timer settings apply to each peer connection. The `neighbor timers` command configures the times on a specified peer connection.

- Keepalive time: period between the transmission of consecutive keepalive messages.
- Hold time: period the switch waits for a keepalive or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The `no timers bgp` and `default timers bgp` commands return the time settings to their default values by removing the `timers bgp` command from `running-config`.

- keepalive: 60 seconds
- hold time: 180 seconds

Command Mode
   Router-BGP Configuration

Command Syntax
```
timers bgp keep_alive hold_time
no timers bgp
default timers bgp
```

Parameters
- `keep_alive`   keepalive period, in seconds. Values include
  - 0   keepalive messages are not sent
  - 1 to 3600  keepalive time (seconds).
- `hold_time`   hold time. Values include
  - 0   peering is not disabled by timeout expiry; keepalive packets are not sent.
  - 3 to 7200  hold time (seconds).

Example
- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds.

```bash
switch(config)#router bgp 9
switch(config-router-bgp)#timers bgp 30 90
switch(config-router-bgp)#
```
**update wait-for-convergence**

The **update wait-for-convergence** command disables FIB update and route advertisement when the BGP instance is initiated until the BGP convergence state is reached.

The **no update wait-for-convergence** command allows FIB update and route advertisement irrespective of the BGP convergence state.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

- `update wait-for-convergence`
- `no update wait-for-convergence`
- `default update wait-for-convergence`

**Related Commands**

- `clear ip bgp` removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.
- `bgp convergence slow-peer time` configures the BGP convergence idle peer timeout value.
- `bgp convergence time` configures the BGP convergence timeout value.
- `show bgp convergence` displays information about the BGP convergence state; and other statistics about the BGP instance in either the specified VRF or all VRFs.

**Guidelines**

The initiation of BGP instance includes the following scenarios:

- The BGP instance starts for the first time after a switch is reloaded
- The BGP instance restarts
- All sessions are cleared by using the `clear ip bgp *` command

Configuration changes made by using this command are effective from the next initiation of BGP instance.

**Example**

- This command disables FIB update and route advertisement when the BGP instance is initiated until the BGP convergence state is reached.

```
switch(config)#router bgp 9
switch(config-router-bgp)#update wait-for-convergence
switch(config-router-bgp)#
```
vrf

The vrf command places the switch in BGP VRF configuration mode for the specified VRF. Commands issued in this mode will override global BGP configuration for the specified VRF.

**Command Mode**

Router-BGP Configuration

**Command Syntax**

```
vrf vrf_instance
```

**Parameters**

- `vrf_instance` VRF to be configured.

**Example**

- These commands place the switch in BGP VRF configuration mode for VRF “purple.”

```
switch(config)#router bgp 9
switch(config-router-bgp)#vrf purple
switch(config-router-bgp-vrf-purple)#
```