



Configuring Advanced BGP

This chapter contains the following topics:

- [About Advanced BGP, on page 2](#)
- [Prerequisites for Advanced BGP, on page 10](#)
- [Guidelines and Limitations for Advanced BGP, on page 11](#)
- [Default Settings, on page 12](#)
- [Configuring Advanced BGP, on page 13](#)
- [Configuring BGP Additional Paths, on page 25](#)
- [Configuring eBGP, on page 28](#)
- [Configuring AS Confederations, on page 30](#)
- [Configuring Route Reflector, on page 31](#)
- [Configuring Next-Hops on Reflected Routes Using an Outbound Route-Map, on page 33](#)
- [Configuring Route Dampening, on page 35](#)
- [Configuring Load Sharing and ECMP, on page 36](#)
- [Configuring Maximum Prefixes, on page 36](#)
- [Configuring DSCP, on page 37](#)
- [Configuring Dynamic Capability, on page 37](#)
- [Configuring Aggregate Addresses, on page 38](#)
- [Suppressing BGP Routes, on page 39](#)
- [Configuring BGP Conditional Advertisement, on page 39](#)
- [Configuring Route Redistribution, on page 41](#)
- [Advertising the Default Route, on page 43](#)
- [Tuning BGP, on page 44](#)
- [Configuring Multiprotocol BGP, on page 48](#)
- [Configuring BMP, on page 50](#)
- [Configuring a Graceful Restart, on page 52](#)
- [Configuring Virtualization, on page 53](#)
- [Verifying the Advanced BGP Configuration, on page 54](#)
- [Monitoring BGP Statistics, on page 56](#)
- [Configuration Examples, on page 57](#)
- [Related Documents, on page 57](#)
- [Additional References, on page 57](#)

About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Cisco NX-OS supports BGP version 4. BGP version 4 includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices called BGP peers. When connecting to an external organization, the router creates external BGP (eBGP) peering sessions. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

Peer Templates

BGP peer templates allow you to create blocks of common configuration that you can reuse across similar BGP peers. Each block allows you to define a set of attributes that a peer then inherits. You can choose to override some of the inherited attributes as well, making it a very flexible scheme for simplifying the repetitive nature of BGP configurations.

Cisco NX-OS implements three types of peer templates:

- The peer-session template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).
- A peer-policy template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Cisco NX-OS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The peer template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Authentication

You can configure authentication for a BGP neighbor session. This authentication method adds an MD5 authentication digest to each TCP segment sent to the neighbor to protect BGP against unauthorized messages and TCP security attacks.



Note The MD5 password must be identical between BGP peers.

Route Policies and Resetting BGP Sessions

You can associate a route policy to a BGP peer. Route policies use route maps to control or modify the routes that BGP recognizes. You can configure a route policy for inbound or outbound route updates. The route policies can match on different criteria, such as a prefix or AS_path attribute, and selectively accept or deny the routes. Route policies can also modify the path attributes.

When you change a route policy that is applied to a BGP peer, you must reset the BGP sessions for that peer. Cisco NX-OS supports the following three mechanisms to reset BGP peering sessions:

- **Hard reset**—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.
- **Soft reconfiguration inbound**—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes that are received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cisco NX-OS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.
- **Route Refresh**—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Cisco NX-OS automatically sends an outbound route refresh of prefixes to the peer.
- BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session. Route refresh is the preferred option and enabled by default.



Note BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features. See [Configuring Route Policy Manager](#), for more information on route maps.

eBGP

External BGP (eBGP) allows you to connect BGP peers from different autonomous systems to exchange routing updates. Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet.

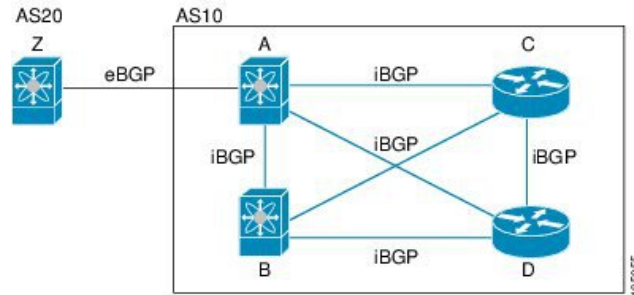
Typically eBGP peerings need to be over directly connected interfaces so that convergence will be faster when the interface goes down.

iBGP

Internal BGP (iBGP) allows you to connect BGP peers within the same autonomous system. You can use iBGP for multihomed BGP networks (networks that have more than one connection to the same external autonomous system).

The figure shows an iBGP network within a larger BGP network.

Figure 1: iBGP Network



iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

You should use loopback interfaces for establishing iBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue. See the [Configuring eBGP, on page 28](#) section for information on multihop, fast external fallovers, and limiting the size of the AS_path attribute.



Note You should configure a separate interior gateway protocol in the iBGP network.

Capabilities Negotiation

A BGP speaker can learn about BGP extensions that are supported by a peer by using the capabilities negotiation feature. Capabilities negotiation allows BGP to use only the set of features supported by both BGP peers on a link.

If a BGP peer does not support capabilities negotiation, Cisco NX-OS attempts a new session to the peer without capabilities negotiation if you have configured the address family as IPv4. Any other multiprotocol configuration (such as IPv6) requires capabilities negotiation.

Route Dampening

Route dampening is a BGP feature that minimizes the propagation of flapping routes across an internetwork. A route flaps when it alternates between the available and unavailable states in rapid succession.

For example, consider a network with three BGP autonomous systems: AS1, AS2, and AS3. Suppose that a route in AS1 flaps (it becomes unavailable). Without route dampening, AS1 sends a withdraw message to AS2. AS2 propagates the withdrawal message to AS3. When the flapping route reappears, AS1 sends an advertisement message to AS2, which sends the advertisement to AS3. If the route repeatedly becomes unavailable, and then available, AS1 sends many withdrawal and advertisement messages that propagate through the other autonomous systems.

Route dampening can minimize flapping. Suppose that the route flaps. AS2 (in which route dampening is enabled) assigns the route a penalty of 1000. AS2 continues to advertise the status of the route to neighbors. Each time that the route flaps, AS2 adds to the penalty value. When the route flaps so often that the penalty exceeds a configurable suppression limit, AS2 stops advertising the route, regardless of how many times that it flaps. The route is now dampened.

The penalty placed on the route decays until the reuse limit is reached. At that time, AS2 advertises the route again. When the reuse limit is at 50 percent, AS2 removes the dampening information for the route.



Note The router does not apply a penalty to a resetting BGP peer when route dampening is enabled, even though the peer reset withdraws the route.

Load Sharing and Multipath

BGP can install multiple equal-cost eBGP or iBGP paths into the routing table to reach the same destination prefix. Traffic to the destination prefix is then shared across all the installed paths.

The BGP best-path algorithm considers the paths as equal-cost paths if the following attributes are identical:

- Weight
- Local preference
- AS_path
- Origin code
- Multi-exit discriminator (MED)
- IGP cost to the BGP next hop

BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers. For more information, see the [BGP Additional Paths, on page 5](#) section.



Note Paths that are received from different autonomous system confederations are considered as equal-cost paths if the external AS_path values and the other attributes are identical.



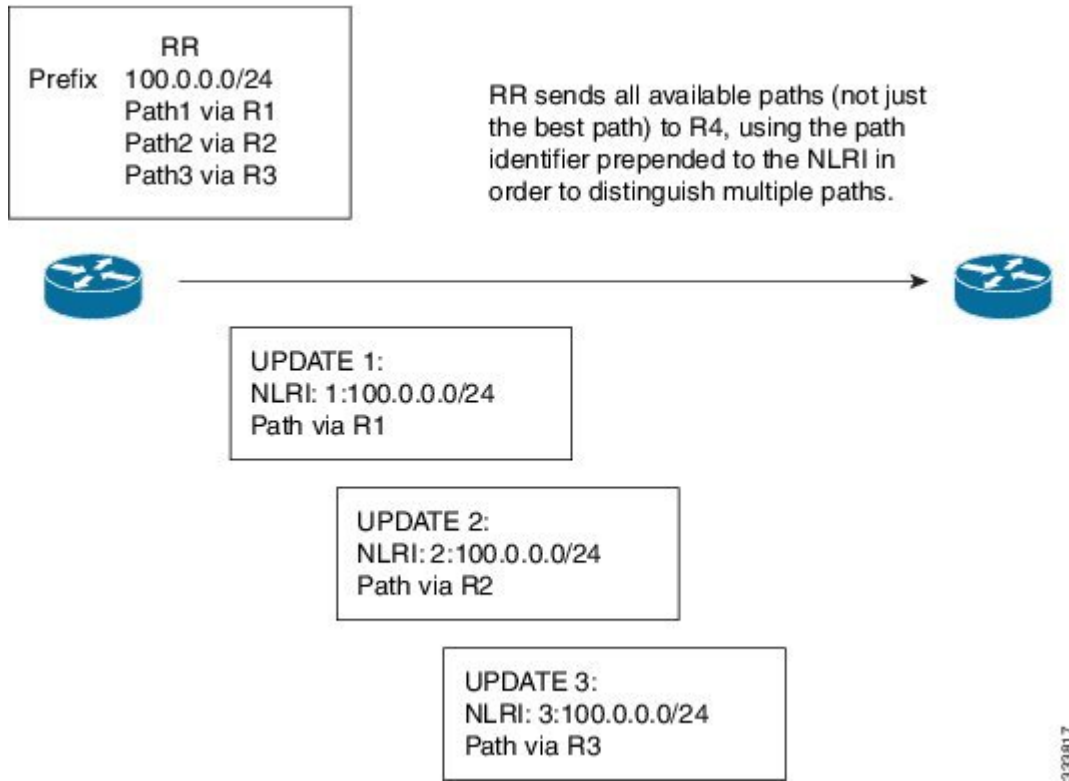
Note When iBGP multipath has a route reflector, and the route reflector advertises the selected best path to its peers, the next hop for the path is not modified.

BGP Additional Paths

Only one BGP best path is advertised, and the BGP speaker accepts only one path for a given prefix from a given peer. If a BGP speaker receives multiple paths for the same prefix within the same session, it uses the most recent advertisement.

BGP supports the additional paths feature, which allows the BGP speaker to propagate and accept multiple paths for the same prefix without the new paths replacing any previous ones. This feature allows BGP speaker peers to negotiate whether they support advertising and receiving multiple paths per prefix and advertising such paths. A special 4-byte path ID is added to the Network Layer Reachability Information (NLRI) to differentiate multiple paths for the same prefix sent across a peer session. The following figure illustrates the BGP additional paths capability.

Figure 2: BGP Route Advertisement with the Additional Paths Capability



For information on configuring BGP additional paths, see the [Configuring BGP Additional Paths, on page 25](#) section.

Route Aggregation

You can configure aggregate addresses. Route aggregation simplifies route tables by replacing a number of more specific addresses with an address that represents all the specific addresses. For example, you can replace these three more specific addresses, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one aggregate address, 10.1.0.0/16.

Aggregate prefixes are present in the BGP route table so that fewer routes are advertised.



Note Cisco NX-OS does not support automatic route aggregation.

Route aggregation can lead to forwarding loops. To avoid this problem, when BGP generates an advertisement for an aggregate address, it automatically installs a summary discard route for that aggregate address in the local routing table. BGP sets the administrative distance of the summary discard to 220 and sets the route type to discard. BGP does not use discard routes for next-hop resolution.

BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether a prefix exists in the BGP table. This feature is useful, for example, in multihomed networks, in which you

want BGP to advertise some prefixes to one of the providers only if information from the other provider is not present.

Consider an example network with three BGP autonomous systems: AS1, AS2, and AS3, where AS1 and AS3 connect to the Internet and to AS2. Without conditional advertisement, AS2 propagates all routes to both AS1 and AS3. With conditional advertisement, you can configure AS2 to advertise certain routes to AS3 only if routes from AS1 do not exist (if, for example, the link to AS1 fails).

BGP conditional advertisement adds an exist or not-exist test to each route that matches the configured route map. See the [Configuring BGP Conditional Advertisement, on page 39](#) section for more information.

BGP Next-Hop Address Tracking

BGP monitors the next-hop address of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. BGP next-hop address tracking speeds up this next-hop reachability test by triggering the verification process when routes change in the Routing Information Base (RIB) that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when the next-hop information changes (event-driven notifications). BGP is notified when any of the following events occurs:

- The next hop becomes unreachable.
- The next hop becomes reachable.
- The fully recurred Interior Gateway Protocol (IGP) metric to the next hop changes.
- The first hop IP address or first hop interface changes.
- The next hop becomes connected.
- The next hop becomes unconnected.
- The next hop becomes a local address.
- The next hop becomes a nonlocal address.



Note Reachability and recurred metric events trigger a best-path recalculation.

Event notifications from the RIB are classified as critical and noncritical. Notifications for critical and noncritical events are sent in separate batches. However, a noncritical event is sent with the critical events if the noncritical event is pending and there is a request to read the critical events.

- Critical events are related to next-hop reachability, such as the loss of next hops resulting in a switchover to a different path. A change in the IGP metric for a next hop resulting in a switchover to a different path can also be considered a critical event.
- Non-critical events are related to next hops being added without affecting the best path or changing the IGP metric to a single next hop.

See the [Configuring BGP Next-Hop Address Tracking, on page 23](#) section for more information.

Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into BGP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. See [Configuring Route Policy Manager](#), for more information.

You can use route maps to override the default behavior in both scenarios, but be careful when doing so. Incorrect use of route maps can result in network loops. The following examples show how to use route maps to change the default behavior.

You can change the default behavior for scenario 1 by modifying the route map as follows:

```
route-map foo permit 10
  match route-type internal
router ospf 1
  redistribute bgp 100 route-map foo
```

Similarly, you can change the default behavior for scenario 2 by modifying the route map as follows:

```
route-map foo deny 10
  match route-type internal
router ospf 1
  vrf bar
  redistribute bgp 100 route-map foo
```

BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4 and IPv6. BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules.

BFD for BGP is supported on eBGP peers and iBGP single-hop peers. Configure the **update-source** option in neighbor configuration mode for iBGP single-hop peers using BFD.

See the *Cisco Nexus 3400-S NX-OS Interfaces Configuration Guide* for more information.

Tuning BGP

You can modify the default behavior of BGP through BGP timers and by adjusting the best-path algorithm.

BGP Timers

BGP uses different types of timers for neighbor session and global protocol events. Each established session has a minimum of two timers for sending periodic keepalive messages and for timing out sessions when peer keepalives do not arrive within the expected time. In addition, there are other timers for handling specific features. Typically, you configure these timers in seconds. The timers include a random adjustment so that the same timers on different BGP peers trigger at different times.

Tuning the Best-Path Algorithm

You can modify the default behavior of the best-path algorithm through optional configuration parameters, including changing how the algorithm handles the multi-exit discriminator (MED) attribute and the router ID.

Multiprotocol BGP

BGP on Cisco NX-OS supports multiple address families. Multiprotocol BGP (MP-BGP) carries different sets of routes depending on the address family. For example, BGP can carry one set of routes for IPv4 unicast

routing, one set of routes for IPv4 multicast routing, and one set of routes for IPv6 multicast routing. You can use MP-BGP for reverse-path forwarding (RPF) checks in IP multicast networks.



Note Because Multicast BGP does not propagate multicast state information, you need a multicast protocol, such as Protocol Independent Multicast (PIM).

Use the router address-family and neighbor address-family configuration modes to support multiprotocol BGP configurations. MP-BGP maintains separate RIBs for each configured address family, such as a unicast RIB and a multicast RIB for BGP.

A multiprotocol BGP network is backward compatible but BGP peers that do not support multiprotocol extensions cannot forward routing information, such as address family identifier information, that the multiprotocol extensions carry.

RFC 5549

BGP supports RFC 5549, which allows an IPv4 prefix to be carried over an IPv6 next hop. Because BGP is running on every hop and all routers are capable of forwarding IPv4 and IPv6 traffic, there is no need to support IPv6 tunnels between any routers. BGP installs IPv4 over an IPv6 route to the Unicast Route Information Base (URIB).

BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) monitors BGP updates and peer statistics and is supported for Cisco Nexus 3400-S platform switches.

Using this protocol, the BGP speaker connects to external BMP servers and sends them information regarding BGP events. A maximum of two BMP servers can be configured in a BGP speaker, and each BGP peer can be configured for monitoring by all or a subset of the BMP servers. The BGP speaker does not accept any information from the BMP server.

Graceful Restart and High Availability

Cisco NX-OS supports nonstop forwarding and graceful restart for BGP.

You can use nonstop forwarding (NSF) for BGP to forward data packets along known routes in the Forward Information Base (FIB) while the BGP routing protocol information is being restored following a failover. With NSF, BGP peers do not experience routing flaps. During a failover, the data traffic is forwarded through intelligent modules while the standby supervisor becomes active.

If a Cisco NX-OS router experiences a cold reboot, the network does not forward traffic to the router and removes the router from the network topology. In this scenario, BGP experiences a nongraceful restart and removes all routes. When Cisco NX-OS applies the startup configuration, BGP reestablishes peering sessions and relearns the routes.

A Cisco NX-OS router that has dual supervisors can experience a stateful supervisor switchover. During the switchover, BGP uses nonstop forwarding to forward traffic based on the information in the FIB, and the system is not removed from the network topology. A router whose neighbor is restarting is referred to as a "helper." After the switchover, a graceful restart operation begins. When it is in progress, both routers reestablish their neighbor relationship and exchange their BGP routes. The helper continues to forward prefixes pointing to the restarting peer, and the restarting router continues to forward traffic to peers even though those neighbor relationships are restarting. When the restarting router has all route updates from all BGP peers that are graceful restart capable, the graceful restart is complete, and BGP informs the neighbors that it is operational again.

When a router detects that a graceful restart operation is in progress, both routers exchange their topology tables. When the router has route updates from all BGP peers, it removes all the stale routes and runs the best-path algorithm on the updated routes.

After the switchover, Cisco NX-OS applies the running configuration, and BGP informs the neighbors that it is operational again.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

With the additional BGP paths feature, if the number of paths advertised for a given prefix is the same before and after restart, the choice of path ID guarantees the final state and removal of stale paths. If fewer paths are advertised for a given prefix after a restart, stale paths can occur on the graceful restart helper peer.

Low Memory Handling

BGP reacts to low memory for the following conditions:

- Minor alert—BGP does not establish any new eBGP peers. BGP continues to establish new iBGP peers and confederate peers. Established peers remain, but reset peers are not re-established.
- Severe alert—BGP shuts down select established eBGP peers every two minutes until the memory alert becomes minor. For each eBGP peer, BGP calculates the ratio of total number of paths that are received to the number of paths that are selected as best paths. BGP selects the peers with the highest ratio to be shut down to reduce memory usage. Clear a shutdown eBGP peer before you bring the eBGP peer back up to avoid oscillation.



Note You can exempt important eBGP peers from this selection process.

- Critical alert—BGP gracefully shuts down all the established peers. You must clear a shutdown BGP peer before you can bring the BGP peer back up.

See the [Tuning BGP , on page 8](#) section for more information on how to exempt a BGP peer from a shutdown due to a low memory condition.

Virtualization Support

You can configure one BGP instance. BGP supports virtual routing and forwarding (VRF) instances.

Prerequisites for Advanced BGP

Advanced BGP has the following prerequisites:

- You must enable BGP. See [Enabling BGP](#).
- You should have a valid router ID configured on the system.
- You must have an autonomous system number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must have reachability (such as an Interior Gateway Protocol [IGP], a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.

- You must explicitly configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Advanced BGP

Advanced BGP has the following configuration guidelines and limitations:

- Prefix peering operates only in passive TCP mode. It accepts incoming connections from remote peers if the peer address falls within the prefix.
- The dynamic autonomous system number prefix peer configuration overrides the individual autonomous system number configuration that is inherited from a BGP template.
- If you configure a dynamic autonomous system number for prefix peers in an autonomous system confederation, BGP establishes sessions with only the autonomous system numbers in the local confederation.
- BGP sessions that are created through a dynamic autonomous system number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes that are received and system resources used.
- Configure the update source to establish a session with eBGP multihop sessions.
- Specify a BGP route map if you configure a redistribution.
- Configure the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, the network might experience session flaps.
- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an extra deny statement into the route map.
- Cisco NX-OS does not support multi-hop BFD. BFD for BGP has the following limitations:
 - BFD is supported only for eBGP peers and iBGP single-hop peers.
 - To enable BFD for iBGP single-hop peers, you must configure the update-source option on the physical interface.
 - BFD is not supported for multi-hop iBGP peers and multi-hop eBGP peers.
 - BGP supports prefix-based peers, but BFD is not supported for prefix-based peers.
- The following guidelines and limitations apply to the **remove-private-as** command:
 - It applies only to eBGP peers.
 - It can be configured only in neighbor configuration mode and not in neighbor-address-family mode.
 - If the AS-path includes both private and public autonomous system numbers, the private autonomous system numbers are not removed.
 - If the AS-path contains the autonomous system number of the eBGP neighbor, the private autonomous system numbers are not removed.

- Private autonomous system numbers are removed only if all autonomous system numbers in that AS-path belong to a private autonomous system number range. Private autonomous system numbers are not removed if a peer's autonomous system number or a non-private autonomous system number is found in the AS-path segment.
- If you use the **aggregate-address** command to configure aggregate addresses and the **suppress-fib-pending** command to suppress BGP routes, lossless traffic for aggregates cannot be ensured on BGP or system triggers.
- When you enable FIB suppression on the switch and route programming fails in the hardware, BGP advertises routes that are not programmed locally in the hardware.
- If you disable a command in the neighbor, template peer, template peer-session, or template peer-policy configuration mode (and the **inherit peer** or **inherit peer-session** command is present), you must use the **default** keyword to return the command to its default state. For example, to disable the **update-source loopback 0** command from the running configuration, you must enter the **default update-source loopback 0** command.
- When next-hop-self is configured for route-reflector clients, the route reflector advertises routes to its clients with itself as the next hop.
- The following guidelines and limitations apply to weighted ECMP:
 - Cisco Nexus 3400-S switches support weighted ECMP.
 - Weighted ECMP is supported only for the IPv4 address family.
 - BGP uses the Link Bandwidth EXTCOMM defined in the draft-ietf-idr-link-bandwidth-06.txt to implement the weighted ECMP feature.
 - BGP accepts the Link Bandwidth EXTCOMM from both iBGP and eBGP peers.

Default Settings

The table lists the default settings for advanced BGP parameters.

| Parameters | Default |
|----------------------|-------------|
| BGP feature | Disabled |
| BGP additional paths | Disabled |
| Keep alive interval | 60 seconds |
| Hold timer | 180 seconds |
| Dynamic capability | Enabled |

Configuring Advanced BGP

Enabling IP Forward on an Interface

To use RFC 5549, you must configure at least one IPv4 address. If you do not want to configure an IPv4 address, you must enable the IP forward feature to use RFC 5549.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)# | Enters interface configuration mode. |
| Step 3 | ip forward Example: switch(config-if)# ip forward | Allows IPv4 traffic on the interface even when there is no IP address configuration on that interface. |
| Step 4 | (Optional) copy running-config startup-config Example: switch(config-if)# copy running-config startup-config | Saves this configuration change. |

Configuring BGP Session Templates

You can use BGP session templates to simplify the BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Before you begin

Enable BGP. See [Enabling BGP](#).



Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 switch(config-router)# | Enables BGP and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | template peer-session <i>template-name</i> Example: switch(config-router)# template peer-session BaseSession switch(config-router-stmp)# | Enters peer-session template configuration mode. |
| Step 4 | (Optional) password <i>number password</i> Example: switch(config-router-stmp)# password 0 test | Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES). |
| Step 5 | (Optional) timers <i>keepalive hold</i> Example: switch(config-router-stmp)# timers 30 90 | Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180. |
| Step 6 | exit Example: switch(config-router-stmp)# exit switch(config-router)# | Exits peer-session template configuration mode. |
| Step 7 | neighbor <i>ip-address remote-as as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)# | Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address. |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 8 | inherit peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor) # inherit peer-session BaseSession switch(config-router-neighbor) #</pre> | Applies a peer-session template to the peer. |
| Step 9 | (Optional) description <i>text</i> Example: <pre>switch(config-router-neighbor) # description Peer Router A switch(config-router-neighbor) #</pre> | Adds a description for the neighbor. |
| Step 10 | (Optional) show bgp peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor) # show bgp peer-session BaseSession</pre> | Displays the peer-policy template. |
| Step 11 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor) # copy running-config startup-config</pre> | Saves this configuration change. Use the show bgp neighbor command to see the template applied. |

Example

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

Cisco NX-OS evaluates multiple peer policies for an address family using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.



Note Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 3000 Series NX-OS N3K Mode Command Reference*, for details on all commands available in the template.

Before you begin

You must enable BGP (see the [Enabling BGP](#) section).



Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: switch# configure terminal | Enters configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 switch(config-router)# | Enables BGP and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | template peer-session <i>template-name</i> Example: switch(config-router)# template peer-policy BasePolicy switch(config-router-ptmp)# | Creates a peer-policy template. |
| Step 4 | (Optional) advertise-active-only Example: switch(config-router-ptmp)# advertise-active-only | Advertises only active routes to the peer. |
| Step 5 | (Optional) maximum-prefix <i>number</i> Example: switch(config-router-ptmp)# maximum-prefix 20 | Sets the maximum number of prefixes allowed from this peer. |
| Step 6 | exit Example: | Exits peer-policy template configuration mode. |

| | Command or Action | Purpose |
|----------------|---|--|
| | <code>switch(config-router-ptmp) # exit</code> <code>switch(config-router) #</code> | |
| Step 7 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <code>switch(config-router) # neighbor</code> <code>192.168.1.2 remote-as 65535</code> <code>switch(config-router-neighbor) #</code> | Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 8 | address-family {ipv4 ipv6} {multicast unicast} Example: <code>switch(config-router-neighbor) #</code> <code>address-family ipv4 unicast</code> <code>switch(config-router-neighbor-af) #</code> | Enters global address family configuration mode for the address family specified. |
| Step 9 | inherit peer-policy <i>template-name</i> preference Example: <code>switch(config-router-neighbor-af) #</code> <code>inherit peer-policy BasePolicy 1</code> | Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy. |
| Step 10 | (Optional) show bgp peer-policy <i>template-name</i> Example: <code>switch(config-router-neighbor-af) # show</code> <code>bgp peer-policy BasePolicy</code> | Displays the peer-policy template. |
| Step 11 | (Optional) copy running-config startup-config Example: <code>switch(config-router-neighbor-af) # copy</code> <code>running-config startup-config</code> | Saves this configuration change. Use the show bgp neighbor command to see the template applied. |

Example

This example shows how to configure a BGP peer-policy template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes that are configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

Before you begin

Enable BGP. See [Enabling BGP](#).



Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | configure terminal Example: switch# configure terminal | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | template peer <i>template-name</i> Example: switch(config-router)# template peer BasePeer | Enters peer template configuration mode. |
| Step 4 | (Optional) inherit peer-session <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer-session BaseSession | Adds a peer-session template to the peer template. |
| Step 5 | (Optional) address-family { ipv4 ipv6 } { multicast unicast } Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af) | Configures the global address family configuration mode for the specified address family. |

| | Command or Action | Purpose |
|----------------|---|---|
| Step 6 | (Optional) inherit peer-policy <i>template-name</i> Example: switch(config-router-neighbor-af) # inherit peer-policy BasePolicy 1 | Applies a peer-policy template to the neighbor address family configuration. |
| Step 7 | exit Example: switch(config-router-neighbor-af) # exit | Exits BGP neighbor address family configuration mode. |
| Step 8 | (Optional) timers <i>keepalive hold</i> Example: switch(config-router-neighbor) # timers 45 100 | Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession. |
| Step 9 | exit Example: switch(config-router-neighbor) # exit | Exits BGP neighbor configuration mode. |
| Step 10 | neighbor <i>ip-address remote-as as-number</i> Example: switch(config-router) # neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor) # | Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 11 | inherit peer <i>template-name</i> Example: switch(config-router-neighbor) # inherit peer BasePeer | Inherits the peer template. |
| Step 12 | (Optional) timers <i>keepalive hold</i> Example: switch(config-router-neighbor) # timers 60 120 | Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template. |
| Step 13 | (Optional) show bgp peer-template <i>template-name</i> Example: switch(config-router-neighbor) # show bgp peer-template BasePeer | Displays the peer template. |
| Step 14 | (Optional) copy running-config startup-config Example: switch(config-router-neighbor) # copy running-config startup-config | Saves this configuration change. Use the show bgp neighbor command to see the template applied. |

Example

This example shows how to configure a BGP peer template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring Prefix Peering

BGP supports the definition of a set of peers using a prefix for both IPv4 and IPv6. This feature allows you to not have to add each neighbor to the configuration.

When defining a prefix peering, you must specify the remote AS number with the prefix. BGP accepts any peer that connects from that prefix and autonomous system if the prefix peering does not exceed the configured maximum peers allowed.

When a BGP peer that is part of a prefix peering disconnects, Cisco NX-OS holds its peer structures for a defined prefix peer timeout value. An established peer can reset and reconnect without danger of being blocked because other peers have consumed all slots for that prefix peering.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <p>timers prefix-peer-timeout <i>value</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# timers prefix-peer-timeout 120</pre> | <p>Configures the BGP prefix peering timeout value in router configuration mode. The range is from 0 to 1200 seconds. The default value is 30.</p> <p>Note For prefix peers, set the prefix peer timeout to be greater than the configured graceful restart timer. If the prefix peer timeout is greater than the graceful restart timer, a peer's route is retained during its restart. If the prefix peer timeout is less than the graceful restart timer, the peer's route is purged by the prefix peer timeout, which may occur before the restart is complete.</p> |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 2 | maximum-peers <i>value</i> Example: <pre>switch(config-router-neighbor) # maximum-peers 120</pre> | Configures the maximum number of peers for this prefix peering in neighbor configuration mode. The range is from 1 to 1000. |

Example

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch(config)# router bgp 65536
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65536
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

Use the **show bgp ipv4 unicast neighbors** command to show the details of the configuration for that prefix peering with a list of the currently accepted instances and the counts of active, maximum concurrent, and total accepted peers.

Configuring BGP Authentication

You can configure BGP to authenticate route updates from peers using MD5 digests.

To configure BGP to use MD5 digests, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | password {0 3 7} <i>string</i> Example: <pre>switch(config-router-neighbor) # password BGPpassword</pre> | Configures an MD5 password for BGP neighbor sessions. |

Resetting a BGP Session

If you modify a route policy for BGP, you must reset the associated BGP peer sessions. If the BGP peers do not support route refresh, you can configure a soft reconfiguration for inbound policy changes. Cisco NX-OS automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | soft-reconfiguration inbound Example: <pre>switch(config-router-neighbor-af) # soft-reconfiguration inbound</pre> | Enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| Step 2 | (Optional) clear bgp {ipv4 ipv6} {unicast multicast} ip-address soft {in out} Example: <pre>switch# clear bgp ip unicast 192.0.2.1 soft in</pre> | Resets the BGP session without tearing down the TCP session. |
| Step 3 | clear bgp {ipv4 ipv6} {unicast multicast} ip-address soft (in out) Example: <pre>switch# clear bgp ip unicast 192.0.2.1 soft in</pre> | Resets the BGP session without tearing down the TCP session. |

Modifying the Next-Hop Address

You can modify the next-hop address used in a route advertisement in the following ways:

- Disable next-hop calculation and use the local BGP speaker address as the next-hop address.
- Set the next-hop address as a third-party address. Use this feature in situations where the original next-hop address is on the same subnet as the peer that the route is being sent to. Using this feature saves an extra hop during forwarding.

To modify the next-hop address, use the following commands in address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | next-hop-self Example: <pre>switch(config-router-neighbor-af) # next-hop-self</pre> | Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| Step 2 | next-hop-third-party Example: <pre>switch(config-router-neighbor-af) # next-hop-third-party</pre> | Sets the next-hop address as a third-party address. Use this command for single-hop eBGP peers that do not have next-hop-self configured. |

Configuring BGP Next-Hop Address Tracking

BGP next-hop address tracking is enabled by default and cannot be disabled.

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking.

To modify the BGP next-hop address tracking, use the following commands in address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | nexthop trigger-delay {critical non-critical} <i>milliseconds</i> Example: <pre>switch(config-router-af)# nexthop trigger-delay critical 5000</pre> | Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000. |

Configuring Next-Hop Filtering

BGP next-hop filtering allows you to specify that when a next-hop address is checked with the RIB, the underlying route for that next-hop address is passed through the route map. If the route map rejects the route, the next-hop address is treated as unreachable.

BGP marks all next hops that are rejected by the route policy as invalid and does not calculate the best path for the routes that use the invalid next-hop address.

To configure BGP next-hop filtering, use the following command in address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | nexthop route-map <i>name</i> Example: <pre>switch(config-router-af)# nexthop route-map nextHopLimits</pre> | Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters. |

Shrinking Next-Hop Groups When A Session Goes Down

You can configure BGP to shrink ECMP groups in an accelerated way when a session goes down.

This feature applies to the following BGP path failure events:

- Any single or multiple Layer 3 link failures
- Line card failures
- BFD failure detections for BGP neighbors
- Administrative shutdown of BGP neighbors (using the shutdown command)

The accelerated handling of the first two events (Layer 3 link failures and line card failures) is enabled by default and does not require a configuration command to be enabled.

To configure the accelerated handling of the last two events, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | neighbor-down fib-accelerate Example: <pre>switch(config-router)# neighbor-down fib-accelerate</pre> | Withdraws the corresponding next hop from all next-hop groups (ECMP groups and single next-hop routes) whenever a BGP session goes down. Note This command applies to both IPv4 and IPv6 routes. |

Disabling Capabilities Negotiation

You can disable capabilities negotiations to interoperate with older BGP peers that do not support capabilities negotiation.

To disable capabilities negotiation, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | dont-capability-negotiate Example: <pre>switch(config-router-neighbor)# dont-capability-negotiate</pre> | Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command. |

Disabling Policy Batching

In BGP deployments where prefixes have unique attributes, BGP tries to identify routes with similar attributes to bundle in the same BGP update message. To avoid the overhead of this additional BGP processing, you can disable batching.

Cisco recommends that you disable policy batching for BGP deployments that have a large number of routes with unique next hops.

To disable policy batching, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | disable-policy-batching Example: | Disables the batching evaluation of prefix advertisements to all peers. |

| | Command or Action | Purpose |
|--|---|---------|
| | switch(config-router)# disable-policy-batching | |

Configuring BGP Additional Paths

BGP supports sending and receiving multiple paths per prefix and advertising such paths.

Advertising the Capability of Sending and Receiving Additional Paths

You can configure BGP to advertise the capability of sending and receiving additional paths to and from the BGP peers. To do so, use the following commands in neighbor address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | [no] capability additional-paths send [disable] Example: switch(config-router-neighbor-af)# capability additional-paths send | Advertises the capability to send additional paths to the BGP peer. The disable option disables the advertising capability of sending additional paths. The no form of this command disables the capability of sending additional paths. |
| Step 2 | [no] capability additional-paths receive [disable] Example: switch(config-router-neighbor-af)# capability additional-paths receive | Advertises the capability to receive additional paths from the BGP peer. The disable option disables the advertising capability of receiving additional paths. The no form of this command disables the capability of receiving additional paths. |
| Step 3 | show bgp neighbor Example: switch(config-router-neighbor-af)# show bgp neighbor | Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer. |

Example

This example shows how to configure BGP to advertise the capability to send and receive additional paths to and from the BGP peer:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)# capability additional-paths receive
```

Configuring the Sending and Receiving of Additional Paths

You can configure the capability of sending and receiving additional paths to and from the BGP peers. To do so, use the following commands in address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | [no] additional-paths send Example: <pre>switch(config-router-af)# additional-paths send</pre> | Enables the send capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The no form of this command disables the send capability. |
| Step 2 | [no] additional-paths receive Example: <pre>switch(config-router-af)# additional-paths receive</pre> | Enables the receive capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The no form of this command disables the receive capability. |
| Step 3 | show bgp neighbor Example: <pre>switch(config-router-af)# show bgp neighbor</pre> | Displays whether the local peer as advertised the additional paths send or receive capability to the remote peer. |

Example

This example shows how to enable the additional paths send and receive capability for all neighbors under the specified address family for which this capability has not been disabled:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

Configuring Advertised Paths

You can specify the paths that are advertised for BGP. To do so, use the following commands in route-map configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | [no] set ip next-hop unchanged Example: | Specifies and unchanged next-hop IP address. |

| | Command or Action | Purpose |
|---------------|--|--|
| | switch(config-route-map)# set ip next-hop unchanged | |
| Step 2 | <p>[no] set path-selection { all backup best2 multipaths } advertise</p> <p>Example:</p> <pre>switch(config-route-map)# set path-selection all advertise</pre> | <p>Specifies that all paths be advertised for a given prefix. You can use one of the following options:</p> <ul style="list-style-type: none"> • all—Advertises all available valid paths. • backup—Advertises paths marked as backup paths. This option requires that backup paths be enabled using the additional-path install backup command. • best2—Advertises the second best path, which is the best path of the remaining available paths, except the already calculated best path. • multipaths—Advertises all multipaths. This option requires that multipaths be enabled using the maximum-paths command. <p>Note If there are no multipaths, the backup and best2 options are the same. If there are multipaths, best2 is the first path on the list of multipaths while backup is the best path of all available paths, except the calculated best path and multipaths.</p> <p>The no form of this command specifies that only the best path be advertised.</p> |
| Step 3 | <p>show bgp {ipv4 ipv6} unicast [ip-address ipv6-prefix] [vrf vrf-name]</p> <p>Example:</p> <pre>switch(config-route-map)# show bgp ipv4 unicast</pre> | <p>Displays the path ID for the additional paths of a prefix and advertisement information for these paths.</p> |

Example

This example show how to specify that all paths be advertised for the prefix list p1:

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

Configuring Additional Path Selection

You can configure the capability for selecting additional paths for a prefix. To do so, use the following commands in address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <p>[no] additional-paths selection route-map <i>map-name</i></p> <p>Example:</p> <pre>switch(config-router-af)# additional paths selection route-map map1</pre> | <p>Configures the capability of selecting additional paths for a prefix.</p> <p>The no form of this command disables the additional paths selection capability.</p> |
| Step 2 | <p>show bgp {ipv4 ipv6} unicast [<i>ip-address</i> <i>ipv6-prefix</i>] [vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>switch(config-router-af)# show bgp ipv4 unicast</pre> | <p>Displays the path ID for the additional paths of a prefix and advertisement information for these paths.</p> |

Example

This example shows how to configure additional paths selection under the specified address family:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

Configuring eBGP

Disabling eBGP Single-Hop Checking

You can configure eBGP to disable checking whether a single-hop eBGP peer is directly connected to the local router. Use this option for configuring a single-hop loopback eBGP session between directly connected switches.

To disable checking whether or not a single-hop eBGP peer is directly connected, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <p>disable-connected-check</p> <p>Example:</p> | <p>Disables checking whether or not a single-hop eBGP peer is directly connected. You must</p> |

| | Command or Action | Purpose |
|--|---|---|
| | <code>switch(config-router-neighbor) # disable-connected-check</code> | manually reset the BGP sessions after using this command. |

Configuring eBGP Multihop

You can configure the eBGP time-to-live (TTL) value to support eBGP multihop. In some situations, an eBGP peer is not directly connected to another eBGP peer and requires multiple hops to reach the remote eBGP peer. You can configure the eBGP TTL value for a neighbor session to allow these multihop sessions.



Note This configuration is not supported for BGP interface peering.

To configure eBGP multihop, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | ebgp-multihop <i>ttl-value</i> Example: <code>switch(config-router-neighbor) # ebgp-multihop 5</code> | Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command. |

Disabling a Fast External Fallover

By default, the Cisco NX-OS device supports fast external fallover for neighbors in all VRFs and address families (IPv4 or IPv6). Typically, when a BGP router loses connectivity to a directly connected eBGP peer, BGP triggers a fast external fallover by resetting the eBGP session to the peer. You can disable this fast external fallover to limit the instability caused by link flaps.

To disable fast external fallover, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | no fast-external-fallover Example: <code>switch(config-router) # no fast-external-fallover</code> | Disables a fast external fallover for eBGP peers. This command is enabled by default. |

Limiting the AS-path Attribute

You can configure eBGP to discard routes that have a high number of AS numbers in the AS-path attribute.

To discard routes that have a high number of AS numbers in the AS-path attribute, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | maxas-limit <i>number</i> Example: switch(config-router)# maxas-limit 50 | Discards eBGP routes that have a number of AS-path segments that exceed the specified limit. The range is from 1 to 2000. |

Configuring Local AS Support

The local-AS feature allows a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

This feature can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation subautonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | local-as <i>number</i> [no-prepend [replace-as [dual-as]]] Example: switch(config-router-neighbor)# local-as 1.1 | Configures eBGP to prepend the local AS <i>number</i> to the AS_PATH attribute. The AS <i>number</i> can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. |

Example

This example shows how to configure local AS support on a VRF:

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# vrf test
switch(config-router-vrf)# local-as 1
switch(config-router-vrf)# show running-config bgp
```

Configuring AS Confederations

To configure an AS confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems within the AS confederation look like a single autonomous system with the confederation identifier as the autonomous system number.

To configure a BGP confederation identifier, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | confederation identifier <i>as-number</i> Example: <pre>switch(config-router)# confederation identifier 4000</pre> | In router configuration mode, this command configures a BGP confederation identifier. The command triggers an automatic notification and session reset for the BGP neighbor sessions. |
| Step 2 | bgp confederation peers <i>as-number</i> [<i>as-number2...</i>] Example: <pre>switch(config-router)# bgp confederation peers 5 33 44</pre> | In router configuration mode, this command configures the autonomous systems that belong to the AS confederation. The command specifies a list of autonomous systems that belong to the confederation and it triggers an automatic notification and session reset for the BGP neighbor sessions. |

Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

Before you begin

You must enable BGP.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal</pre> | Enters global configuration mode. |
| Step 2 | router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | cluster-id <i>cluster-id</i> Example: | Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This |

| | Command or Action | Purpose |
|----------------|---|--|
| | <code>switch(config-router)# cluster-id 192.0.2.1</code> | command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| Step 4 | address-family {ipv4 ipv6} {unicast multicast} Example: <code>switch(config-router)# address-family ipv4 unicast</code> <code>switch(config-router-af)#</code> | Enters router address family configuration mode for the specified address family. |
| Step 5 | (Optional) client-to-client reflection Example: <code>switch(config-router-af)# client-to-client reflection</code> | Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| Step 6 | exit Example: <code>switch(config-router-af)# exit</code> <code>switch(config-router)#</code> | Exits router address configuration mode. |
| Step 7 | neighbor ip-address remote-as as-number Example: <code>switch(config-router)# neighbor 192.0.2.10 remote-as 65535</code> <code>switch(config-router-neighbor)#</code> | Configures the IP address and AS number for a remote BGP peer. |
| Step 8 | address-family {ipv4 ipv6} {unicast multicast} Example: <code>switch(config-router-neighbor)# address-family ipv4 unicast</code> <code>switch(config-router-neighbor-af)#</code> | Enters neighbor address family configuration mode for the unicast IPv4 address family. |
| Step 9 | route-reflector-client Example: <code>switch(config-router-neighbor-af)# route-reflector-client</code> | Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions. |
| Step 10 | (Optional) show bgp {ipv4 ipv6} {unicast multicast} neighbors Example: <code>switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors</code> | Displays the BGP peers. |
| Step 11 | (Optional) copy running-config startup-config Example: | Saves this configuration change. |

| | Command or Action | Purpose |
|--|--|---------|
| | switch(config-router-neighbor-af)# copy running-config startup-config | |

Example

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Next-Hops on Reflected Routes Using an Outbound Route-Map

You can change the next-hop on reflected routes on a BGP route reflector using an outbound route-map. You can configure the outbound route-map to specify the peer's local address as the next-hop address.



Note The **next-hop-self** command does not enable this functionality for routes being reflected to clients by a route reflector. This functionality can only be enabled using an outbound route-map.

Before you begin

Enable BGP. See [Enabling BGP](#).

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

You must enter the **set next-hop** command to configure an address family-specific next-hop address. For example, for the IPv6 address family, you must enter the **set ipv6 next-hop peer-address** command.

- When setting IPv4 next-hops using route-maps—If **set ip next-hop peer-address** matches the route-map, the next-hop is set to the peer's local address. If no next-hop is set in the route-map, the next-hop is set to the one stored in the path.
- When setting IPv6 next-hops using route-maps—If **set ipv6 next-hop peer-address** matches the route-map, the next-hop is set as follows:
 - For IPv6 peers, the next-hop is set to the peer's local IPv6 address.
 - For IPv4 peers, if **update-source** is configured, the next-hop is set to the source interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set.
 - For IPv4 peers, if **update-source** is not configured, the next-hop is set to the outgoing interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set.

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | router bgp <i>as-number</i> Example: switch(config)# router bgp 200 switch(config-router)# | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.0.2.12 remote-as 200 switch(config-router-neighbor)# | Configures the IP address and autonomous system number for a remote BGP peer. |
| Step 4 | (Optional) update-source <i>interface number</i> Example: switch(config-router-neighbor)# update-source loopback 300 | Specifies and updates the source of the BGP session. |
| Step 5 | address-family {<i>ipv4</i> <i>ipv6</i>} {unicast multicast} Example: switch(config-router-neighbor)# address-family <i>ipv4</i> unicast switch(config-router-neighbor-af)# | Enters router address family configuration mode for the specified address family. |
| Step 6 | route-reflector-client Example: switch(config-router-neighbor-af)# route-reflector-client | Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions. |
| Step 7 | route-map <i>map-name</i> out Example: switch(config-router-neighbor-af)# route-map setrrnh out | Applies the configured BGP policy to outgoing routes. |
| Step 8 | (Optional) show bgp {<i>ipv4</i> <i>ipv6</i>} {unicast multicast} [<i>ip-address</i> <i>ipv6-prefix</i>] route-map <i>map-name</i> [<i>vrf vrf-name</i>] Example: switch(config-router-neighbor-af)# show bgp <i>ipv4</i> unicast route-map setrrnh | Displays the BGP routes that match the route map. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| Step 9 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor-af)# copy running-config startup-config</pre> | Saves this configuration change. |

Example

This example shows how to configure the next-hop on reflected routes on a BGP route reflector using an outbound route-map:

```
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ipv6 address 2001::a0c:1a65/64
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# route-map setrrnhv6 permit 10
switch(config-route-map)# set ipv6 next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnhv6 out
```

Configuring Route Dampening

You can configure route dampening to minimize route flaps propagating through your iBGP network.

To configure route dampening, use the following command in address-family or VRF address family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | dampening [<i>{half-life reuse-limit suppress-limit max-suppress-time route-map map-name}</i>] Example: <pre>switch(config-router-af)# dampening route-map bgpDamp</pre> | Disables capabilities negotiation. The parameter values are as follows: <ul style="list-style-type: none"> • <i>half-life</i>—The range is from 1 to 45. • <i>reuse-limit</i>—The range is from 1 to 20000. |

| | Command or Action | Purpose |
|--|-------------------|---|
| | | <ul style="list-style-type: none"> • <i>suppress-limit</i>—The range is from 1 to 20000. • <i>max-suppress-time</i>—The range is from 1 to 255. |

Configuring Load Sharing and ECMP

You can configure the maximum number of paths that BGP adds to the route table for equal-cost multipath (ECMP) load balancing.

To configure the maximum number of paths, use the following command in router address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | maximum-paths [ibgp] <i>maxpaths</i> Example: <pre>switch(config-router-af) # maximum-paths 8</pre> | Configures the maximum number of equal-cost paths for load sharing. The default is 1. |

Configuring Maximum Prefixes

You can configure the maximum number of prefixes that BGP can receive from a BGP peer. If the number of prefixes exceeds this value, you can optionally configure BGP to generate a warning message or tear down the BGP session to the peer.

To configure the maximum allowed prefixes for a BGP peer, use the following command in neighbor address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | maximum-prefix <i>maximum</i> [<i>threshold</i>] [restart <i>time</i> warning-only] Example: <pre>switch(config-router-neighbor-af) # maximum-prefix 12</pre> | Configures the maximum number of prefixes from a peer. The parameter ranges are as follows: <ul style="list-style-type: none"> • <i>maximum</i>—The range is from 1 to 300000. • <i>threshold</i>—The range is from 1 to 100 percent. The default is 75 percent. • <i>time</i>—The range is from 1 to 65535 minutes. |

| | Command or Action | Purpose |
|--|-------------------|--|
| | | This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix is exceeded. |

Configuring DSCP

You can configure a differentiated services code point (DSCP) for a neighbor. You can specify a DSCP value for locally originated packets for IPv4 or IPv6.

To configure the DSCP value, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | <p>dscp <i>dscp_value</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# dscp 63</pre> <p>Below is an example of the corresponding show command:</p> <pre>show ipv6 bgp neighbors BGP neighbor is 10.1.1.1, remote AS 0, unknown link, Peer index 4 BGP version 4, remote router ID 0.0.0.0 BGP state = Idle, down for 00:13:34, retry in 0.000000 DSCP (DiffServ CodePoint): 0 Last read never, hold time = 180, keepalive interval is 60 seconds</pre> | <p>Sets the differentiated services code point (DSCP) value for the neighbor. The DSCP value can be a number from 0 to 63, or it can be one of the following keywords: ef, af11, af12, af13, af21, af22, af23, af31, af32, af33, af41, af42, af43, cs1, cs2, cs3, cs4, cs5, cs6, or cs7.</p> <p>The default value is cs6.</p> |

Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | <p>dynamic-capability</p> <p>Example:</p> <pre>switch(config-router-neighbor)# dynamic-capability</pre> | <p>Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.</p> |

Configuring Aggregate Addresses

You can configure aggregate address entries in the BGP route table.

You can also set BGP attributes for the more-specific routes matched in an aggregate-route.

To configure an aggregate address, use the following command in router address-family configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | <p>aggregate-address <i>ip-prefix/length</i> [as-set] [summary-only] [advertise-map <i>map-name</i>] [attribute-map <i>map-name</i>] [suppress-map <i>map-name</i>]</p> <p>Example:</p> <pre>switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set</pre> | <p>Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized:</p> <ul style="list-style-type: none"> • The as-set keyword generates autonomous system set path information and community information from contributing paths. • The summary-only keyword filters all more specific routes from updates. • The advertise-map keyword and argument specify the route map used to select attribute information from selected routes. • The attribute-map keyword and argument specify the route map used to select attribute information from the aggregate. • The suppress-map keyword and argument conditionally filter more specific routes. If you specify the suppress-map option while performing a BGP route aggregation, you can set the community attribute for a BGP route update. This option enables you to set community attributes on the more-specific routes. |
| Step 2 | <p>set community { none additive local-AS no-advertise no-export <i>community-1</i> [<i>community-2...</i>]}]</p> <p>Example:</p> <pre>switch(config-router-af)# aggregate-address 192.0.2.0/8 suppress-map map1 switch(cconfig-router-af)# set community local-AS</pre> | <p>If you specify the suppress-map option while performing a BGP route aggregation, you can set the community attribute for a BGP route update. This option enables you to set community attributes on the more-specific routes.</p> <ul style="list-style-type: none"> • The none keyword specifies no community attribute. |

| | Command or Action | Purpose |
|--|-------------------|--|
| | | <ul style="list-style-type: none"> • The additive keyword specifies to add the aggregated BGP route to the existing community. • The local-AS keyword specifies not to send the aggregated BGP route outside local AS (well-known community). • The no-advertise keyword specifies not to advertise the aggregated BGP route to any peer (well-known community). • The no-export keyword specifies not to export the aggregated BGP route to the next AS (well-known community). |

Suppressing BGP Routes

You can configure Cisco NX-OS to advertise newly learned BGP routes only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware. After the routes are programmed, subsequent changes to these routes do not require this hardware-programming check.

To suppress BGP routes, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | suppress-fib-pending Example: <pre>switch(config-router)# suppress-fib-pending</pre> | Suppresses newly learned BGP routes (IPv4 or IPv6) from being advertised to downstream BGP neighbors until the routes have been programmed in the hardware. |

Configuring BGP Conditional Advertisement

You can configure BGP conditional advertisement to limit the routes that BGP propagates. You define the following two route maps:

- **Advertise map**—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.
- **Exist map or nonexist map**—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

If the route does not pass the condition, BGP withdraws the route if it exists in the BGP table.

Before you begin

You must enable BGP. See the [Enabling BGP](#) section.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters configuration mode. |
| Step 2 | router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65534 switch(config-router-neighbor)#</pre> | Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 4 | address-family {<i>ipv4</i> <i>ipv6</i>} {unicast multicast} Example: <pre>switch(config-router-neighbor)# address-family ipv4 multicast switch(config-router-neighbor-af)#</pre> | Enters address family configuration mode. |
| Step 5 | advertise-map <i>adv-map</i> {<i>exist-map</i> <i>exist-rmap</i> <i>non-exist-map</i> <i>nonexist-rmap</i>} Example: <pre>switch(config-router-neighbor-af)# advertise-map advertise exist-map exist</pre> | Configures BGP to conditionally advertise routes based on the two configured route maps: <ul style="list-style-type: none"> • <i>adv-map</i>—Specifies a route map with match statements that the route must pass before BGP passes the route to the next route map. The <i>adv-map</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>exist-rmap</i>—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP advertises the route. The <i>exist-rmap</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>nonexist-rmap</i>—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP advertises the route. The <i>nonexist-rmap</i> is |

| | Command or Action | Purpose |
|---------------|--|---|
| | | a case-sensitive, alphanumeric string up to 63 characters. |
| Step 6 | (Optional) show bgp {ipv4 ipv6} {unicast multicast} neighbors Example: switch(config-router-neighbor-af)# show ip bgp neighbor | Displays information about BGP and the configured conditional advertisement route maps. |
| Step 7 | (Optional) copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config | Saves this configuration change. |

Example

This example shows how to configure BGP conditional advertisement:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

Configuring Route Redistribution

You can configure BGP to accept routing information from another routing protocol and redistribute that information through the BGP network. Optionally, you can assign a default route for redistributed routes.

Before you begin

You must enable BGP.

Procedure

| | Command or Action | Purpose |
|---------------|--|-----------------------------------|
| Step 1 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--|--|
| | switch# configure terminal switch(config)# | |
| Step 2 | router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)# | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | address-family { ipv4 ipv6 } { unicast multicast } Example: switch(config-router)# address-family vpv4 unicast switch(config-router-af)# | Enters address family configuration mode. |
| Step 4 | address-family { ipv4 ipv6 } { unicast multicast } Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)# | Enters address-family configuration mode. |
| Step 5 | redistribute { direct { eigrp isis ospf ospfv3 rip } instance-tag static } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap | Redistributes routes from other protocols into BGP. |
| Step 6 | redistribute { direct { eigrp isis ospf ospfv3 rip } instance-tag static } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap | Redistributes routes from other protocols into BGP. |
| Step 7 | (Optional) default-metric <i>value</i> Example: switch(config-router-af)# default-metric 33 | Generates a default route into BGP. |
| Step 8 | (Optional) copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config | Saves this configuration change. |

Example

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap
switch(config-router-af)# copy running-config startup-config
```

Advertising the Default Route

You can configure BGP to advertise the default route (network 0.0.0.0).

Before you begin

Enable BGP. See [Enabling BGP](#).

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | route-map allow permit Example: switch(config)# route-map allow permit switch(config-route-map)# | Enters router map configuration mode and defines the conditions for redistributing routes. |
| Step 3 | exit Example: switch(config-route-map)# exit switch(config)# | Exits router map configuration mode. |
| Step 4 | ip route ip-address network-mask null null-interface-number Example: switch(config)# ip route 192.0.2.1 255.255.255.0 null 0 | Configures the IP address. |
| Step 5 | router bgp as-number Example: switch(config)# router bgp 65535 switch(config-router)# | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 6 | address-family {ipv4 ipv6} unicast Example: <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre> | Enters address-family configuration mode. |
| Step 7 | default-information originate Example: <pre>switch(config-router-af)# default-information originate</pre> | Advertises the default route. |
| Step 8 | redistribute static route-map allow Example: <pre>switch(config-router-af)# redistribute static route-map allow</pre> | Redistributes the default route. |
| Step 9 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-af)# copy running-config startup-config</pre> | Saves this configuration change. |

Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

To tune BGP, use the following optional commands in router configuration mode:

| Command | Purpose |
|---|---|
| <p>bestpath [always-compare-med as-path multipath-relax compare-routerid cost-community ignore igp-metric ignore med {confed missing-as-worst non-deterministic}]</p> <p>Example:</p> <pre>switch(config-router)# bestpath always-compare-med</pre> | <p>Modifies the best-path algorithm. The optional parameters are as follows:</p> <ul style="list-style-type: none"> • always-compare-med—Compares MED on paths from different autonomous systems. • as-path multipath-relax—Allows load sharing across the providers with different (but equal-length) AS paths. Without this option, the AS paths must be identical for load sharing. • compare-routerid—Compares the router IDs for identical eBGP paths. • cost-community ignore—Ignores the cost community for BGP best-path calculations. • igp-metric ignore—Ignores the Interior Gateway Protocol (IGP) metric for next hop during best-path selection. • med confed—Forces bestpath to do a MED comparison only between paths originated within a confederation. • med missing-as-worst—Treats a missing MED as the highest MED. • med non-deterministic—Does not always pick the best MED path from among the paths from the same autonomous system. |
| <p>enforce-first-as</p> <p>Example:</p> <pre>switch(config-router)# enforce-first-as</pre> | <p>Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.</p> |
| <p>log-neighbor-changes</p> <p>Example:</p> <pre>switch(config-router)# log-neighbor-changes</pre> | <p>Generates a system message when any neighbor changes state.</p> <p>Note To suppress neighbor status change messages for a specific neighbor, you can use the log-neighbor-changes disable command in router address-family configuration mode.</p> |
| <p>router-id <i>id</i></p> <p>Example:</p> <pre>switch(config-router)# router-id 10.165.20.1</pre> | <p>Manually configures the router ID for this BGP speaker.</p> |

| Command | Purpose |
|---|--|
| <p>timers [bestpath-delay <i>delay</i> <i>bgpkeepalive holdtime</i> prefix-peer-timeout <i>timeout</i>]</p> <p>Example:</p> <pre>switch(config-router)# timers bgp 90 270</pre> | <p>Sets BGP timer values. The optional parameters are as follows:</p> <ul style="list-style-type: none"> • <i>delay</i>—Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. • <i>keepalive</i>—BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. • <i>holdtime</i>—BGP session hold time. The range is from 0 to 3600 seconds. The default value is 180. • <i>timeout</i>—Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30. <p>You must manually reset the BGP sessions after configuring this command.</p> |

To tune BGP, use the following optional commands in router address-family configuration mode:

| Command | Purpose |
|--|---|
| <p>distance <i>ebgp-distance</i> <i>ibgp-distance</i> <i>local-distance</i></p> <p>Example:</p> <pre>switch(config-router-af)# distance 20 100 200</pre> | <p>Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows:</p> <ul style="list-style-type: none"> • <i>ebgp-distance</i>—20. • <i>ibgp-distance</i>—200. • <i>local-distance</i>—220. Local-distance is the administrative distance used for aggregate discard routes when they are installed in the RIB. <p>After you enter the value for the external administrative distance, you must enter the value for the administrative distance for the internal routes or/and the value for the administrative distance for the local routes depending on your requirement; so that the internal/local routes are also considered in the route administration.</p> |
| <p>log-neighbor-changes [disable]</p> <p>Example:</p> <pre>switch(config-router-af)# log-neighbor-changes disable</pre> | <p>Generates a system message when this specific neighbor changes state.</p> <p>The disable option suppresses neighbor status changes messages for this specific neighbor.</p> |

To tune BGP, use the following optional commands in neighbor configuration mode:

| Command | Purpose |
|--|--|
| <p>description <i>string</i></p> <p>Example:</p> <pre>switch(config-router-neighbor) # description main site</pre> | <p>Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.</p> |
| <p>low-memory exempt</p> <p>Example:</p> <pre>switch(config-router-neighbor) # low-memory exempt</pre> | <p>Exempts this BGP neighbor from a possible shutdown due to a low memory condition.</p> |
| <p>transport connection-mode passive</p> <p>Example:</p> <pre>switch(config-router-neighbor) # transport connection-mode passive</pre> | <p>Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.</p> |
| <p>[no default] remove-private-as [all replace-as]</p> <p>Example:</p> <pre>switch(config-router-neighbor) # remove-private-as</pre> | <p>Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.</p> <p>The optional parameters are as follows:</p> <ul style="list-style-type: none"> • no—Disables the command. • default—Moves the command to its default mode. • all—Removes all private-as numbers from the AS-path value. • replace-as—Replaces all private AS numbers with the replace-as AS-path value. |
| <p>update-source <i>interface-type number</i></p> <p>Example:</p> <pre>switch(config-router-neighbor) # update-source ethernet 2/1</pre> | <p>Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions. Single-hop iBGP peers support fast external fallover when update-source is configured.</p> |

To tune BGP, use the following optional commands in neighbor address-family configuration mode:

| Command | Purpose |
|---|---|
| <p>allowas in</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # allowas in</pre> | <p>Allows routes that have their own AS in the AS path to be installed in the BRIB.</p> |

| Command | Purpose |
|---|---|
| default-originate [route-map <i>map-name</i>] Example: <pre>switch(config-router-neighbor-af) # default-originate</pre> | Generates a default route to the BGP peer. |
| disable-peer-as-check Example: <pre>switch(config-router-neighbor-af) # disable-peer-as-check</pre> | Disables peer AS-number checking while the device advertises routes learned from one node to another node in the same AS path. |
| filter-list <i>list-name</i> { in out } Example: <pre>switch(config-router-neighbor-af) # filter-list BGPFilter in</pre> | Applies an AS_path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| prefix-list <i>list-name</i> { in out } Example: <pre>switch(config-router-neighbor-af) # prefix-list PrefixFilter in</pre> | Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| send-community Example: <pre>switch(config-router-neighbor-af) # send-community</pre> | Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| send-community extended Example: <pre>switch(config-router-neighbor-af) # send-community extended</pre> | Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| suppress-inactive Example: <pre>switch(config-router-neighbor-af) # suppress-inactive</pre> | Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. |

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 and IPv6 unicast and multicast routes.

Before you begin

You must enable BGP.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)# | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65534 switch(config-router-neighbor)# | Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 4 | address-family {<i>ipv4</i> <i>ipv6</i>} {unicast multicast} Example: switch(config-router-neighbor)# address-family ipv4 multicast switch(config-router-neighbor-af)# | Enters address family configuration mode. |
| Step 5 | (Optional) copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config | Saves this configuration change. |

Example

This example shows how to enable advertising and receiving IPv4 and IPv6 routes for multicast RPF for a neighbor:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65536
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BMP

You can configure BMP on the device.

Before you begin

Enable BGP. See [Enabling BGP](#).

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | router bgp as-number Example: <pre>switch(config)# router bgp 200 switch(config-router)#</pre> | Enters BGP mode and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | bmp server server-number Example: <pre>switch(config-router)# bmp server 1</pre> | Configures the BMP server to which BGP should send information. The server number is used as a key. Note You can configure up to two BMP servers. |
| Step 4 | address ip-address port-number port-number Example: <pre>switch(config-router)# address 10.1.1.1 port-number 2000</pre> | Configures the IPv4 or IPv6 address of the host and the port number on which the BMP speaker connects to the BMP server. |
| Step 5 | description string Example: <pre>switch(config-router)# description BMPserver1</pre> | Configures the BMP server description. You can enter up to 256 alphanumeric characters. |
| Step 6 | initial-refresh { skip delay time } Example: <pre>switch(config-router)# initial-refresh delay 100</pre> | Configures the option to send a route refresh when BGP is converged and the BMP server connection is established later. The skip option specifies to not send a route refresh if the BMP server connection comes up later. The delay option specifies the time in seconds after which the route refresh should be sent. |

| | Command or Action | Purpose |
|----------------|--|---|
| | | The range is from 30 through 720 seconds, and the default value is 30 seconds. |
| Step 7 | initial-delay <i>time</i> Example: switch(config-router)# initial-delay 120 | Configures the delay after which a connection is attempted to the BMP server. The range is 30–720 seconds, and the default value is 45 seconds. |
| Step 8 | stats-reporting-period <i>time</i> Example: switch(config-router)# stats-reporting-period 50 | Configures the time interval in which the BMP server receives the statistics report from BGP neighbors. The range is 30–720 seconds, and the default is disabled. |
| Step 9 | shutdown Example: switch(config-router)# shutdown | Disables the connection to the BMP server. |
| Step 10 | neighbor ip-address Example: switch(config-router)# neighbor 192.168.1.2 switch(config-router-neighbor)# | Enters neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 11 | remote-as <i>as-number</i> Example: switch(config-router-neighbor)# remote-as 65535 | Configures the autonomous system number for a remote BGP peer. |
| Step 12 | bmp-activate-server <i>server-number</i> Example: switch(config-router-neighbor)# bmp-activate-server 1 | Configures the BMP server to which a neighbor's information should be sent. |
| Step 13 | (Optional) show bgp bmp <i>server</i> <i>[server-number] [detail]</i> Example: switch(config-router-neighbor)# show bgp bmp server | Displays BMP server information. |
| Step 14 | (Optional) copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config | Saves this configuration change. |

Configuring a Graceful Restart

You can configure a graceful restart and enable the graceful restart helper feature for BGP.

Before you begin

Enable BGP. See [Enabling BGP](#).

Create the VRFs.

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: <pre>switch# configure terminal switch(config)#</pre> | Enters configuration mode. |
| Step 2 | router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Creates a new BGP process with the configured autonomous system number. |
| Step 3 | graceful-restart Example: <pre>switch(config-router)# graceful-restart</pre> | <p>Enables a graceful restart and the graceful restart helper functionality. This command is enabled by default.</p> <p>This command triggers an automatic notification and session reset for the BGP neighbor sessions.</p> |
| Step 4 | graceful-restart {restart-time <i>time</i> stalepath-time <i>time</i>} Example: <pre>switch(config-router)# graceful-restart restart-time 300</pre> | <p>Configures the graceful restart timers.</p> <p>The optional parameters are as follows:</p> <ul style="list-style-type: none"> • restart-time—Maximum time for a restart sent to the BGP peer. The range is from 1 to 3600 seconds. The default is 120. • stalepath-time—Maximum time that BGP keeps the stale routes from the restarting BGP peer. The range is from 1 to 3600 seconds. The default is 300. <p>This command triggers an automatic notification and session reset for the BGP neighbor sessions.</p> |
| Step 5 | graceful-restart-helper Example: <pre>switch(config-router)# graceful-restart restart-time 300</pre> | <p>Enables the graceful restart helper functionality. Use this command if you have disabled graceful restart but you still want to enable graceful restart helper functionality. This command triggers an automatic notification and session reset for the BGP neighbor sessions.</p> |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| Step 6 | (Optional) show running-config bgp Example: switch(config-router)# show running-config bgp | Displays the BGP configuration. |
| Step 7 | (Optional) copy running-config startup-config Example: switch(config-router)# copy running-config startup-config | Saves this configuration change. |

Example

This example shows how to enable a graceful restart:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# graceful-restart
switch(config-router)# copy running-config startup-config
```

Configuring Virtualization

You can configure one BGP process, create multiple VRFs, and use the same BGP process in each VRF.

Before you begin

You must enable BGP.

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: switch# configure terminal switch(config)# | Enters global configuration mode. |
| Step 2 | vrf context vrf-name Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)# | Creates a new VRF and enters VRF configuration mode. |
| Step 3 | exit Example: switch(config-vrf)# exit switch(config)# | Exits VRF configuration mode. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 4 | router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Creates a new BGP process with the configured autonomous system number. |
| Step 5 | vrf <i>vrf-name</i> Example: <pre>switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#</pre> | Enters the router VRF configuration mode and associates this BGP instance with a VRF. |
| Step 6 | neighbor <i>ip-address remote-as as-number</i> Example: <pre>switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535 switch(config-router--vrf-neighbor)#</pre> | Configures the IP address and autonomous system number for a remote BGP peer. |
| Step 7 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-vrf-neighbor)# copy running-config startup-config</pre> | Saves this configuration change. |

Example

This example shows how to create a VRF and configure the router ID in the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65536
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

To display the BGP configuration, perform one of the following tasks:

| Command | Purpose |
|--|--|
| show bgp all [summary] [vrf <i>vrf-name</i>] | Displays the BGP information for all address families. |
| show bgp convergence [vrf <i>vrf-name</i>] [detail] | Displays the BGP information for all address families. |

| Command | Purpose |
|--|---|
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community { regex <i>expression</i> [community] [no-advertise] [no-export] [no-export-subconfed]} [vrf <i>vrf-name</i>] | Displays the BGP routes that match a BGP community. |
| show bgp [vrf <i>vrf-name</i>] { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community-list <i>list-name</i> [vrf <i>vrf-name</i>] | Displays the BGP routes that match a BGP community list. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity { regex <i>expression</i> generic [non-transitive transitive] <i>aa4:nn</i> [exact-match]} [vrf <i>vrf-name</i>] | Displays the BGP routes that match a BGP extended community. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity-list <i>list-name</i> [exact-match]} [vrf <i>vrf-name</i>] | Displays the BGP routes that match a BGP extended community list. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity-list <i>list-name</i> [exact-match]} [vrf <i>vrf-name</i>] | Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] { dampening dampened-paths [regex <i>expression</i>]} [vrf <i>vrf-name</i>] | Displays the BGP route history paths. |
| show bgp { ipv4 ipv6 vpn4 vpn6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] filter-list <i>list-name</i> [vrf <i>vrf-name</i>] | Displays the information for the BGP filter list. |
| show bgp { ipv4 ipv6 vpn4 vpn6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] neighbors [<i>ip-address</i> <i>ipv6-prefix</i>] [vrf <i>vrf-name</i>] | Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] { nexthop nexthop-database } [vrf <i>vrf-name</i>] | Displays the information for the BGP route next hop. |
| show bgp paths | Displays the BGP path information. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] policy <i>name</i> [vrf <i>vrf-name</i>] | Displays the BGP policy information. Use the clear bgp policy command to clear the policy information. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] prefix-list <i>list-name</i> [vrf <i>vrf-name</i>] | Displays the BGP routes that match the prefix list. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] received-paths [vrf <i>vrf-name</i>] | Displays the BGP paths stored for soft reconfiguration. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] regex <i>expression</i> [vrf <i>vrf-name</i>] | Displays the BGP routes that match the AS_path regular expression. |
| show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] route-map <i>map-name</i> [vrf <i>vrf-name</i>] | Displays the BGP routes that match the route map. |

| Command | Purpose |
|---|--|
| <code>show bgp peer-policy name [vrf vrf-name]</code> | Displays the information about BGP peer policies. |
| <code>show bgp peer-session name [vrf vrf-name]</code> | Displays the information about BGP peer sessions. |
| <code>show bgp peer-template name [vrf vrf-name]</code> | Displays the information about BGP peer templates. Use the clear bgp peer-template command to clear all neighbors in a peer template. |
| <code>show bgp process</code> | Displays the BGP process information. |
| <code>show ip route ip-address detail vrf all i bw</code> | Displays the link bandwidth EXTCOMM fields. bw:xx (such as bw:40) in the output indicates that BGP peers are sending BGP extended attributes with the bandwidth (for weighted ECMP). |
| <code>show {ipv4 ipv6} bgp options</code> | Displays the BGP status and configuration information. |
| <code>show {ipv4 ipv6} mbgp options</code> | Displays the BGP status and configuration information. |
| <code>show running-configuration bgp</code> | Displays the current running BGP configuration. |

Monitoring BGP Statistics

To display BGP statistics, use the following commands:

| Command | Purpose |
|---|---|
| <code>show bgp {ipv4 ipv6} {unicast multicast} [ip-address ipv6-prefix] flap-statistics [vrf vrf-name]</code> | Displays the BGP route flap statistics. Use the clear bgp flap-statistics command to clear these statistics. |
| <code>show bgp {ipv4 ipv6} unicast injected-routes</code> | Displays injected routes in the routing table. |
| <code>show bgp sessions [vrf vrf-name]</code> | Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics. |
| <code>show bgp statistics</code> | Displays the BGP statistics. |

Configuration Examples

This example shows how to configure MD5 authentication for prefix-based neighbors:

```
template peer BasePeer-V6
  description BasePeer-V6
  password 3 f4200cfc725bbd28
  transport connection-mode passive
  address-family ipv6 unicast
template peer BasePeer-V4
  bfd
  description BasePeer-V4
  password 3 f4200cfc725bbd28
  address-family ipv4 unicast
--
neighbor fc00::10:3:11:0/127 remote-as 65006
  inherit peer BasePeer-V6
neighbor 10.3.11.0/31 remote-as 65006
  inherit peer BasePeer-V4
```

This example shows how to enable neighbor status change messages globally and suppress them for a specific neighbor:

```
router bgp 65100
  log-neighbor-changes
  neighbor 209.165.201.1 remote-as 65535
    description test
    address-family ipv4 unicast
    soft-reconfiguration inbound
    disable log-neighbor-changes
```

Related Documents

| Related Topic | Document Title |
|------------------|---|
| BGP CLI commands | <i>Cisco Nexus 3000 Series NX-OS N3K Mode Command Reference</i> |

Additional References

For additional information related to implementing BGP, see the following sections:

MIBs

| MIBs | MIBs Link |
|---------------------|--|
| MIBs related to BGP | To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus3000/Nexus3000MIBSupportList.html |

