



Configuring Advanced BGP

This chapter contains the following sections:

- [About Advanced BGP, on page 2](#)
- [Prerequisites for Advanced BGP, on page 11](#)
- [Guidelines and Limitations for Advanced BGP, on page 11](#)
- [Default Settings, on page 12](#)
- [Configuring BGP Session Templates, on page 13](#)
- [Configuring BGP Peer-Policy Templates, on page 15](#)
- [Configuring BGP Peer Templates, on page 17](#)
- [Configuring Prefix Peering, on page 19](#)
- [Configuring BGP Authentication, on page 20](#)
- [Resetting a BGP Session, on page 21](#)
- [Modifying the Next-Hop Address, on page 21](#)
- [Configuring BGP Next-Hop Address Tracking, on page 22](#)
- [Configuring Next-Hop Filtering, on page 22](#)
- [Configuring Next-Hop Resolution via Default Route, on page 23](#)
- [Controlling Reflected Routes Through Next-Hop-Self, on page 23](#)
- [Shrinking Next-Hop Groups When A Session Goes Down, on page 24](#)
- [Disabling Capabilities Negotiation, on page 24](#)
- [Disabling Policy Batching, on page 25](#)
- [Configuring BGP Additional Paths, on page 25](#)
- [Configuring eBGP, on page 29](#)
- [Configuring AS Confederations, on page 31](#)
- [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 Maximum Prefixes, on page 36](#)
- [Configuring DSCP, on page 36](#)
- [Configuring Dynamic Capability, on page 37](#)
- [Configuring Aggregate Addresses, on page 37](#)
- [Suppressing BGP Routes, on page 38](#)
- [Configuring BGP Conditional Advertisement, on page 39](#)
- [Configuring Route Redistribution, on page 41](#)
- [Advertising the Default Route, on page 42](#)

- [Configuring BGP Attribute Filtering and Error Handling, on page 43](#)
- [Tuning BGP, on page 46](#)
- [Configuring Policy-Based Administrative Distance, on page 50](#)
- [Configuring Multiprotocol BGP, on page 51](#)
- [Configuring BMP, on page 52](#)
- [About BGP Graceful Shutdown, on page 54](#)
- [Graceful Shutdown Aware and Activate, on page 55](#)
- [Graceful Shutdown Contexts, on page 55](#)
- [Graceful Shutdown with Route Maps, on page 56](#)
- [Guidelines and Limitations, on page 57](#)
- [Graceful Shutdown Task Overview, on page 57](#)
- [Configuring Graceful Shutdown on a Link, on page 58](#)
- [Filtering BGP Routes and Setting Local Preference Based On GRACEFUL_SHUTDOWN Communities, on page 59](#)
- [Configuring Graceful Shutdown for All BGP Neighbors, on page 61](#)
- [Controlling the Preference for All Routes with the GRACEFUL_SHUTDOWN Community, on page 62](#)
- [Preventing Sending the GRACEFUL_SHUTDOWN Community to a Peer, on page 62](#)
- [Displaying Graceful Shutdown Information, on page 63](#)
- [Graceful Shutdown Configuration Examples, on page 64](#)
- [Configuring a Graceful Restart, on page 66](#)
- [Verifying the Advanced BGP Configuration, on page 68](#)
- [Monitoring BGP Statistics, on page 69](#)
- [Related Topics, on page 70](#)
- [Additional References, on page 70](#)

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 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 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.

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.

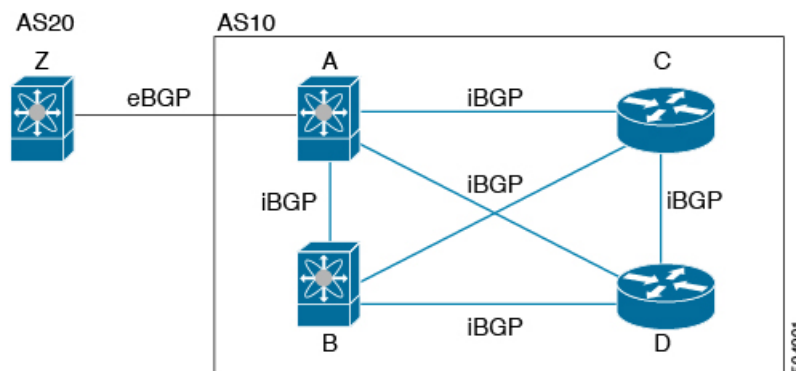
Typically eBGP peerings need to be over directly connected interfaces so that convergence will be faster when the interface goes down.

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 29](#) section for information on multihop, fast external fallovers, and limiting the size of the AS_path attribute.



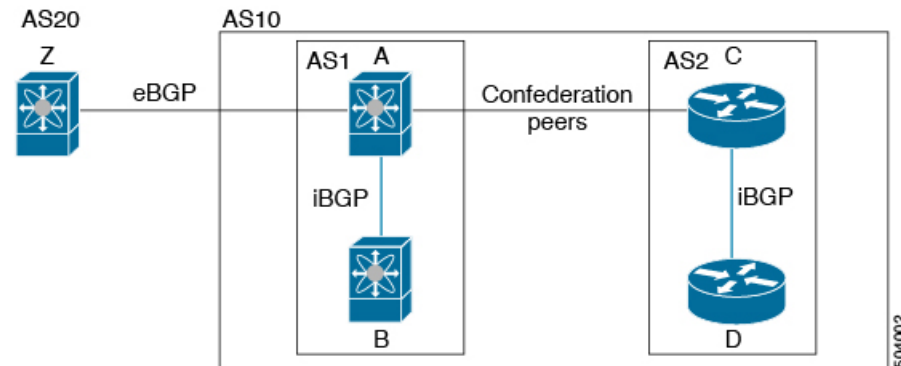
Note You should configure a separate interior gateway protocol in the iBGP network.

AS Confederations

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple subautonomous systems and grouping them into a single confederation. A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the same confederation.

The figure shows the BGP network, split into two subautonomous systems and one confederation.

Figure 2: AS Confederation



In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system.

Route Reflector

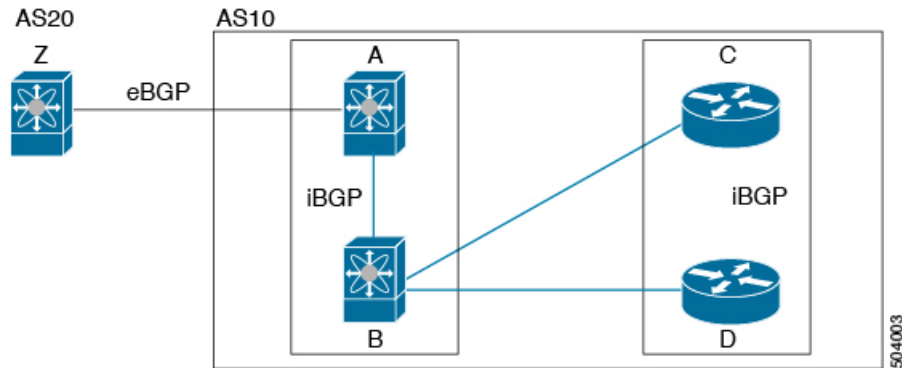
You can alternately reduce the iBGP mesh by using a route reflector configuration where route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

The figure shows a simple iBGP configuration with four meshed iBGP speakers (routers A, B, C, and D). Without route reflectors, when router A receives a route from an external neighbor, it advertises the route to all three iBGP neighbors.

In the figure, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 3: Route Reflector



The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

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.

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.

BGP Best-Path Selection

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 7](#) section.



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



Note When you configure a route reflector for iBGP multipath, 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 4: 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.



Note Cisco Nexus 3550-T hardware does not install ECMP routes.

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.

A summary entry is created in the BGP table when you issue the **aggregate-address** command, but the summary entry is not eligible for advertisement until a subset of the aggregate is found in the table.

BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether or not 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 recursed 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 recursed 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 22](#) 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 the Configuring Route Policy Manager section, for more information.

You can use route maps to override the default behavior in both scenarios, but be careful when doing so as 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
```

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.

Graceful Restart and High Availability

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

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.

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.

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 received to the number of paths selected as best paths. The peers with the highest ratio are selected to be shut down to reduce memory usage. You must clear a shutdown eBGP peer before you can 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 9](#) section for more information on how to exempt a BGP peer from a shutdown due to a low memory condition.

Prerequisites for Advanced BGP

Advanced BGP has the following prerequisites:

- You must enable BGP (see the Enabling BGP section).
- You should have a valid router ID configured on the system.
- You must have an AS 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



Note *Cisco Nexus 3550-T - 10.1(2t) release*, BGP supports only default VRF.

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.
- Configuring the **advertise-maps** command multiple times is not supported.
- The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions that are created through a dynamic AS 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.



Note Only 48 BGP sessions are validated in Cisco Nexus 3550-T.

- 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.
- 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 AS numbers, the private AS numbers are not removed.
 - If the AS-path contains the AS number of the eBGP neighbor, the private AS numbers are not removed.
 - Private AS numbers are removed only if all AS numbers in that AS-path belong to a private AS number range. Private AS numbers are not removed if a peer's AS number or a non-private AS number is found in the AS-path segment.
- 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.

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 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

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: <pre>switch# configure terminal switch(config)#</pre> | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Enables BGP and assigns the autonomous system number to the local BGP speaker. |
| Step 3 | template peer-session <i>template-name</i> Example: <pre>switch(config-router)# template peer-session BaseSession switch(config-router-stmp)#</pre> | Enters peer-session template configuration mode. |
| Step 4 | (Optional) password <i>number password</i> Example: <pre>switch(config-router-stmp)# password 0 test</pre> | Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES). |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 5 | (Optional) timers <i>keepalive hold</i> Example: <pre>switch(config-router-stmp)# timers 30 90</pre> | 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: <pre>switch(config-router-stmp)# exit switch(config-router)#</pre> | Exits peer-session template configuration mode. |
| Step 7 | neighbor <i>ip-address remote-as as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#</pre> | Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| 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.

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. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 3 | template peer-session <i>template-name</i> Example: <pre>switch(config-router)# template peer-policy BasePolicy switch(config-router-ptmp)#</pre> | Creates a peer-policy template. |
| Step 4 | (Optional) advertise-active-only Example: <pre>switch(config-router-ptmp)# advertise-active-only</pre> | Advertises only active routes to the peer. |
| Step 5 | (Optional) maximum-prefix <i>number</i> Example: <pre>switch(config-router-ptmp)# maximum-prefix 20</pre> | Sets the maximum number of prefixes allowed from this peer. |
| Step 6 | exit Example: <pre>switch(config-router-ptmp)# exit switch(config-router)#</pre> | Exits peer-policy template configuration mode. |
| Step 7 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#</pre> | Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address. |
| Step 8 | address-family {<i>ipv4</i>} {<i>unicast</i>} Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre> | Enters global address family configuration mode for the address family specified. |
| Step 9 | inherit peer-policy <i>template-name</i> <i>preference</i> Example: <pre>switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1</pre> | 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: <pre>switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy</pre> | Displays the peer-policy template. |
| Step 11 | (Optional) copy running-config startup-config Example: | Saves this configuration change. Use the show bgp neighbor command to see the template applied. |

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

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 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

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 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. |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 3 | template peer <i>template-name</i> Example: <pre>switch(config-router)# template peer BasePeer</pre> | Enters peer template configuration mode. |
| Step 4 | (Optional) inherit peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor)# inherit peer-session BaseSession</pre> | Adds a peer-session template to the peer template. |
| Step 5 | (Optional) address-family { ipv4 } { unicast } Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)</pre> | Configures the global address family configuration mode for the specified address family. |
| Step 6 | (Optional) inherit peer-policy <i>template-name</i> Example: <pre>switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1</pre> | Applies a peer-policy template to the neighbor address family configuration. |
| Step 7 | exit Example: <pre>switch(config-router-neighbor-af)# exit</pre> | Exits BGP neighbor address family configuration mode. |
| Step 8 | (Optional) timers <i>keepalive hold</i> Example: <pre>switch(config-router-neighbor)# timers 45 100</pre> | Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession. |
| Step 9 | exit Example: <pre>switch(config-router-neighbor)# exit</pre> | Exits BGP neighbor configuration mode. |
| Step 10 | neighbor ip-address remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#</pre> | 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: <pre>switch(config-router-neighbor)# inherit peer BasePeer</pre> | Inherits the peer template. |
| Step 12 | (Optional) timers <i>keepalive hold</i> Example: | Adds the BGP timer values to this neighbor. |

| | Command or Action | Purpose |
|----------------|--|---|
| | <code>switch(config-router-neighbor) # timers 60 120</code> | 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: <code>switch(config-router-neighbor) # show bgp peer-template BasePeer</code> | Displays the peer template. |
| Step 14 | (Optional) copy running-config startup-config Example: <code>switch(config-router-neighbor) # copy 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 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. 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 | timers prefix-peer-timeout <i>value</i> Example: <pre>switch(config-router-neighbor)# timers prefix-peer-timeout 120</pre> | Configures the BGP prefix peering timeout value in router configuration mode. The range is from 0 to 1200 seconds. The default value is 30. 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. |
| 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} string 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} {unicast 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} milliseconds 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. |

Configuring Next-Hop Resolution via Default Route

BGP next-hop resolution allows you to specify if the IP default route is used for BGP next-hop resolution.

To configure BGP next-hop resolution, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | [no] nexthop suppress-default-resolution Example: <pre>switch(config-router) # nexthop suppress-default-resolution</pre> | Prevents resolution of BGP next hop through the IP default route. When this command is enabled: <ul style="list-style-type: none"> The output of the show bgp process detail command includes the following line: Use default route for nexthop resolution: No The output of the show routing clients bgp command includes the following line: Owned rnh will never resolve to 0.0.0.0/0 |

Controlling Reflected Routes Through Next-Hop-Self

NX-OS enables controlling the iBGP routes being sent to a specific peer through the **next-hop-self** [all] arguments. By using these arguments, you can selectively change the next-hop of routes even if the route is reflected.

| Command | Purpose |
|---|---|
| next-hop-self [all] Example: <pre>switch(config-router-af)# next-hop-self all</pre> | <p>Uses the local BGP speaker address as the next-hop address in route updates.</p> <p>The all keyword is optional. If you specify all, all routes are sent to the peer with next-hop-self. If you do not specify all, the next hops of reflected routes are not changed.</p> |

Shrinking Next-Hop Groups 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
- 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> | <p>Withdraws the corresponding next hop from all next-hop groups (single next-hop routes) whenever a BGP session goes down.</p> <p>Note This command applies to both IPv4 routes.</p> |

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: <pre>switch(config-router) # disable-policy-batching</pre> | Disables the batching evaluation of prefix advertisements to all peers. |

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: | Advertises the capability to send additional paths to the BGP peer. The disable option disables the advertising capability of sending additional paths. |

| | Command or Action | Purpose |
|---------------|--|--|
| | <code>switch(config-router-neighbor-af) # capability additional-paths send</code> | The no form of this command disables the capability of sending additional paths. |
| Step 2 | [no] capability additional-paths receive [disable] Example: <code>switch(config-router-neighbor-af) # capability additional-paths receive</code> | 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: <code>switch(config-router-neighbor-af) # show bgp neighbor</code> | 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: <code>switch(config-router-af) # additional-paths send</code> | 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: <code>switch(config-router-af) # additional-paths receive</code> | Enables the receive capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. |

| | Command or Action | Purpose |
|---------------|---|---|
| | | 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: <pre>switch(config-route-map)# set ip next-hop unchanged</pre> | Specifies and unchanged next-hop IP address. |
| Step 2 | [no] set path-selection { all backup best2 } advertise Example: <pre>switch(config-route-map)# set path-selection all advertise</pre> | Specifies that all paths be advertised for a given prefix. You can use one of the following options: <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. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | The no form of this command specifies that only the best path be advertised. |
| Step 3 | show bgp {ipv4 } unicast [<i>ip-address</i>] Example: switch(config-route-map)# show bgp ipv4 unicast | Displays the path ID for the additional paths of a prefix and advertisement information for these paths. |

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

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 | disable-connected-check Example: <pre>switch(config-router-neighbor) # disable-connected-check</pre> | Disables checking whether or not a single-hop eBGP peer is directly connected. You must 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>ttn-value</i> Example: <pre>switch(config-router-neighbor) # ebgp-multihop 5</pre> | 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). 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: <pre>switch(config-router)# no fast-external-fallover</pre> | 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: <pre>switch(config-router)# maxas-limit 50</pre> | 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: | 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 |

| | Command or Action | Purpose |
|--|--|---|
| | <code>switch(config-router-neighbor) # local-as 1.1</code> | 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)# neighbor 10.1.1.1
switch(config-router-neighbor) # local-as 1
switch(config-router-neighbor) # 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: <code>switch(config-router) # confederation identifier 4000</code> | 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: <code>switch(config-router) # bgp confederation peers 5 33 44</code> | 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: switch# configure terminal | 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 | cluster-id <i>cluster-id</i> Example: switch(config-router)# cluster-id 192.0.2.1 | 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 triggers an automatic soft clear or refresh of BGP neighbor sessions. |
| Step 4 | address-family {ipv4} {unicast} Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)# | Enters router address family configuration mode for the specified address family. |
| Step 5 | (Optional) client-to-client reflection Example: switch(config-router-af)# client-to-client reflection | 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: switch(config-router-af)# exit switch(config-router)# | Exits router address configuration mode. |
| Step 7 | neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.0.2.10 remote-as 65535 switch(config-router-neighbor)# | Configures the IP address and AS number for a remote BGP peer. |
| Step 8 | address-family {ipv4} {unicast} Example: | Enters neighbor address family configuration mode for the unicast IPv4 address family. |

| | Command or Action | Purpose |
|----------------|--|--|
| | <pre>switch(config-router-neighbor) # address-family ipv4 unicast switch(config-router-neighbor-af) #</pre> | |
| Step 9 | route-reflector-client Example: <pre>switch(config-router-neighbor-af) # route-reflector-client</pre> | 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} {unicast} neighbors Example: <pre>switch(config-router-neighbor-af) # show bgp ipv4 unicast neighbors</pre> | Displays the BGP peers. |
| Step 11 | (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 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

You must enable BGP (see the Enabling BGP section).

You must enter the **set next-hop** command to configure an address family-specific next-hop address.

- 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.

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 AS 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>unicast</i>} Example: switch(config-router-neighbor)# address-family ipv4 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>unicast</i>} [<i>ip-address</i>] route-map <i>map-name</i> Example: | Displays the BGP routes that match the route map. |

| | Command or Action | Purpose |
|---------------|--|----------------------------------|
| | <pre>switch(config-router-neighbor-af) # show bgp ipv4 unicast route-map setrrnh</pre> | |
| Step 9 | <p>(Optional) copy running-config startup-config</p> <p>Example:</p> <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) # 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) # 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
```

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 configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | <p>dampening [{<i>half-life reuse-limit suppress-limit max-suppress-time</i> route-map map-name}]</p> <p>Example:</p> <pre>switch(config-router-af) # dampening route-map bgpDamp</pre> | <p>Disables capabilities negotiation. The parameter values are as follows:</p> <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. • <i>suppress-limit</i>—The range is from 1 to 20000. • <i>max-suppress-time</i>—The range is from 1 to 255. |

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>] [<i>restart 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. 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.

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

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | dscp <i>dscp_value</i> Example: <pre>switch(config-router-neighbor) # dscp 63</pre> Below is an example of the corresponding show command: <pre>show ipv4 bgp neighbors BGP neighbor is 10.1.1.1, remote AS 0,</pre> | 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 . The default value is cs6 . Note |

| | Command or Action | Purpose |
|--|---|--|
| | <pre>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> | Cisco Nexus 3550-T hardware does not look at the DSCP values in the packets. |

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

Configuring Aggregate Addresses

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

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

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | 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>] Example: <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. |

| | Command or Action | Purpose |
|--|-------------------|--|
| | | <ul style="list-style-type: none"> • 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 either suppress certain more-specific routes from being advertised to its peers, or decide to advertise the more-specific routes with some community attributes set on them, depending upon the suppress-map route-map configuration. A route-map configured with only match clauses will suppress the more-specific routes that satisfy the match criteria. However, if a route-map is configured with match and set clauses, then the routes satisfying the match criteria will be advertised with the appropriate attributes as modified by the route-map. The second option enables you to set community attributes on the more-specific routes. |

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) 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>} {unicast} Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast 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 |

| | Command or Action | Purpose |
|---------------|---|---|
| | | <p>route map. The <i>adv-map</i> is a case-sensitive, alphanumeric string up to 63 characters.</p> <ul style="list-style-type: none"> • <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 a case-sensitive, alphanumeric string up to 63 characters. |
| Step 6 | <p>(Optional) show bgp {ipv4} {unicast} neighbors</p> <p>Example:</p> <pre>switch(config-router-neighbor-af)# show ip bgp neighbor</pre> | Displays information about BGP and the configured conditional advertisement route maps. |
| Step 7 | <p>(Optional) copy running-config startup-config</p> <p>Example:</p> <pre>switch(config-router-neighbor-af)# copy running-config startup-config</pre> | 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.

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 | address-family ipv4 {unicast} Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)# | Enters address family configuration mode. |
| Step 4 | redistribute {direct {ospf} instance-tag static} route-map <i>map-name</i> Example: switch(config-router-af)# redistribute ospf 201 route-map Ospfmap | Redistributes routes from other protocols into BGP. |
| Step 5 | (Optional) default-metric <i>value</i> Example: switch(config-router-af)# default-metric 33 | Generates a default route into BGP. |
| Step 6 | (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 ospf 201 route-map Ospfmap
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

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 global configuration mode. |
| Step 2 | route-map allow permit Example: <pre>switch(config)# route-map allow permit switch(config-route-map)#</pre> | Enters router map configuration mode and defines the conditions for redistributing routes. |
| Step 3 | exit Example: <pre>switch(config-route-map)# exit switch(config)#</pre> | Exits router map configuration mode. |
| Step 4 | ip route <i>ip-address network-mask</i> null <i>null-interface-number</i> Example: <pre>switch(config)# ip route 192.0.2.1 255.255.255.0 null 0</pre> | Configures the IP address. |
| Step 5 | router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65535 switch(config-router)#</pre> | Enters BGP mode and assigns the AS number to the local BGP speaker. |
| Step 6 | address-family {<i>ipv4</i>} unicast Example: <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre> | Enters address-family configuration mode. |

| | Command or Action | Purpose |
|---------------|---|----------------------------------|
| Step 7 | default-information originate Example: <code>switch(config-router-af) # default-information originate</code> | Advertises the default route. |
| Step 8 | redistribute static route-map allow Example: <code>switch(config-router-af) # redistribute static route-map allow</code> | Redistributes the default route. |
| Step 9 | (Optional) copy running-config startup-config Example: <code>switch(config-router-af) # copy running-config startup-config</code> | Saves this configuration change. |

Configuring BGP Attribute Filtering and Error Handling

You can configure BGP attribute filtering and error handling to provide an increased level of security. The following features are available and implemented in the following order:

- **Path attribute treat-as-withdraw:** Allows you to treat-as-withdraw a BGP update from a specific neighbor if the update contains a specified attribute type. The prefixes contained in the update are removed from the routing table.
- **Path attribute discard:** Allows you to remove specific path attributes in a BGP update from a specific neighbor.
- **Enhanced attribute error handling:** Prevents peer sessions from flapping due to a malformed update.

Attribute types 1, 2, 3, 4, 5, 8, 14, 15, and 16 cannot be configured for path attribute treat-as-withdraw and path attribute discard. Attribute type 9 (Originator) and type 10 (Cluster-id) can be configured for eBGP neighbors only.

Treating as Withdraw Path Attributes from a BGP Update Message

To "treat-as-withdraw" BGP updates that contain specific path attributes, use the following command in router neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | [no] path-attribute treat-as-withdraw [value range start end] in Example: <code>switch#(config-router) # neighbor 10.20.30.40</code> | Treats as withdraw any incoming BGP update messages that contain the specified path attribute or range of path attributes and triggers an inbound route refresh to ensure that the routing table is up to date. Any prefixes in a |

| | Command or Action | Purpose |
|--|--|---|
| | <pre>switch(config-router-neighbor)# path-attribute treat-as-withdraw 100 in</pre> <p>Example:</p> <pre>switch#(config-router)# neighbor 10.20.30.40 switch(config-router-neighbor)# path-attribute treat-as-withdraw range 21 255 in</pre> | <p>BGP update that are treat-as-withdraw are removed from the BGP routing table.</p> <p>This command is also supported for BGP template peers and BGP template peer sessions.</p> |

Discarding Path Attributes from a BGP Update Message

To discard BGP updates that contain specific path attributes, use the following command in router neighbor configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <p>[no] path-attribute discard <i>[value range start end]</i> in</p> <p>Example:</p> <pre>switch#(config-router)# neighbor 10.20.30.40 switch(config-router-neighbor)# path-attribute discard 100 in</pre> <p>Example:</p> <pre>switch#(config-router)# neighbor 10.20.30.40 switch(config-router-neighbor)# path-attribute discard range 100 255 in</pre> | <p>Drops specified path attributes in BGP update messages for the specified neighbor and triggers an inbound route refresh to ensure that the routing table is up to date. You can configure a specific attribute or an entire range of unwanted attributes.</p> <p>This command is also supported for BGP template peers and BGP template peer sessions.</p> <p>Note When the same path attribute is configured for both discard and treat-as-withdraw, treat-as-withdraw has a higher priority.</p> |

Enabling or Disabling Enhanced Attribute Error Handling

BGP enhanced attribute error handling is enabled by default but can be disabled. This feature, which complies with RFC 7606, prevents peer sessions from flapping due to a malformed update. The default behavior applies to both eBGP and iBGP peers.

To disable or reenabling enhanced error handling, use the following command in router configuration mode:

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | <p>[no] enhanced-error</p> <p>Example:</p> | <p>Enables or disables BGP enhanced attribute error handling.</p> |

| | Command or Action | Purpose |
|--|--|---------|
| | switch(config)# router bgp 1000 switch(config-router)# enhanced-error | |

Displaying Discarded or Unknown Path Attributes

To display information about discarded or unknown path attributes, perform one of the following tasks:

| Command | Purpose |
|---|--|
| show bgp {ipv4 } unicast path-attribute discard] | Displays all prefixes for which an attribute has been discarded. |
| show bgp {ipv4 } unicast path-attribute unknown] | Displays all prefixes that have an unknown attribute. |
| show bgp {ipv4 } unicast ip-address | Displays the unknown attributes and discarded attributes associated with a prefix. |

The following example shows the prefixes for which an attribute has been discarded:

```
switch# show bgp ipv4 unicast path-attribute discard
Network          Next Hop
1.1.1.1/32        20.1.1.1
1.1.1.2/32        20.1.1.1
1.1.1.3/32        20.1.1.1
```

The following example shows the prefixes that have an unknown attribute:

```
switch# show bgp ipv4 unicast path-attribute unknown
Network          Next Hop
2.2.2.2/32        20.1.1.1
2.2.2.3/32        20.1.1.1
```

The following example shows the unknown attributes and discarded attributes associated with a prefix:

```
switch# show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 6241
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1000
    20.1.1.1 from 20.1.1.1 (20.1.1.1)
      Origin IGP, localpref 100, valid, external, best
    unknown transitive attribute: flag 0xE0 type 0x62 length 0x64
      value 0000 0000 0100 0000 0200 0000 0300 0000
            0400 0000 0500 0000 0600 0000 0700 0000
            0800 0000 0900 0000 0A00 0000 0B00 0000
            0C00 0000 0D00 0000 0E00 0000 0F00 0000
            1000 0000 1100 0000 1200 0000 1300 0000
            1400 0000 1500 0000 1600 0000 1700 0000
            1800 0000
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 20 2019 07:50:43 PST
```

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>Note If BGP computes an ECMP route, it is installed in Cisco Nexus 3550-T hardware as a unipath and a "Failure to install ECMP" warning sys log is generated.</p> | <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> |

| Command | Purpose |
|---|--|
| router-id <i>id</i> Example: <pre>switch(config-router)# router-id 10.165.20.1</pre> | Manually configures the router ID for this BGP speaker. |
| timers [bestpath-delay <i>delay</i> <i>bgpkeepalive holdtime</i> prefix-peer-timeout <i>timeout</i>] Example: <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 |
|--|--|
| distance <i>ebgp-distance ibgp-distance local-distance</i> Example: <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> |

| Command | Purpose |
|--|--|
| log-neighbor-changes [disable] Example: <pre>switch(config-router-af)# log-neighbor-changes disable</pre> | Generates a system message when this specific neighbor changes state. The disable option suppresses neighbor status changes messages for this specific neighbor. |

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

| Command | Purpose |
|---|---|
| description <i>string</i> Example: <pre>switch(config-router-neighbor)# description main site</pre> | Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters. |
| low-memory exempt Example: <pre>switch(config-router-neighbor)# low-memory exempt</pre> | Exempts this BGP neighbor from a possible shutdown due to a low memory condition. |
| transport connection-mode passive Example: <pre>switch(config-router-neighbor)# transport connection-mode passive</pre> | 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. |
| [no default] remove-private-as [all replace-as] Example: <pre>switch(config-router-neighbor)# remove-private-as</pre> | 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. The optional parameters are as follows: <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. See the Guidelines and Limitations for Advanced BGP, on page 11 section for additional information on this command. |

| Command | Purpose |
|--|--|
| update-source <i>interface-type number</i> Example: <pre>switch(config-router-neighbor) # update-source ethernet 1/1</pre> | 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. |

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

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

| Command | Purpose |
|--|--|
| 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. |
| [no default] as-override Example: <pre>switch(config-router-neighbor-af) # as-override</pre> | no - (Optional) Disables the command. default - (Optional) Moves the command to its default mode. as-override - While sending updates to eBGP peer, replaces in the <i>path</i> attribute all occurrences of the peer's AS number with the local AS number. |

Configuring Policy-Based Administrative Distance

You can configure a distance for external BGP (eBGP) and internal BGP (iBGP) routes that match a policy described in the configured route map. The distance configured in the route map is downloaded to the unicast RIB along with the matching routes. BGP uses the best path to determine the administrative distance when downloading next hops in the unicast RIB table. If there is no match or a deny clause in the policy, BGP uses the distance configured in the distance command or the default distance for routes.

The policy-based administrative distance feature is useful when there are two or more different routes to the same destination from two different routing protocols.

Before you begin

You must enable BGP.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# ip prefix-list <i>name seq number permit prefix-length</i> | Creates a prefix list to match IP packets or routes with the permit keyword. |
| Step 3 | switch(config)# route-map <i>map-tag permit sequence-number</i> | Creates a route map and enters route-map configuration mode with the permit keyword. If the match criteria for the route is met in the policy, the packet is policy routed. |
| Step 4 | switch(config-route-map)# match ip address prefix-list <i>prefix-list-name</i> | Matches IPv4 network routes based on a prefix list. The prefix-list name can be any alphanumeric string up to 63 characters. |
| Step 5 | switch(config-route-map)# set distance <i>value1 value2 value3</i> | Specifies the administrative distance for interior BGP (iBGP) or exterior BGP (eBGP) |

| | Command or Action | Purpose |
|----------------|---|---|
| | | <p>routes and BGP routes originated in the local autonomous system. The range is from 1 to 255.</p> <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> |
| Step 6 | switch(config-route-map)# exit | Exits route-map configuration mode. |
| Step 7 | switch(config)# router bgp <i>as-number</i> | Enters BGP mode and assigns the AS number to the local BGP speaker. |
| Step 8 | switch(config-router)# address-family { ipv4 vpn4 } unicast | Enters address family configuration mode. |
| Step 9 | switch(config-router-af)# table-map <i>map-name</i> | <p>Configures the selective administrative distance for a route map for BGP routes before forwarding them to the RIB table. The table-map name can be any alphanumeric string up to 63 characters.</p> <p>Note You can also configure the table-map command under the VRF address-family configuration mode.</p> |
| Step 10 | (Optional) switch(config-router-af)# show forwarding distribution | Displays forwarding information distribution. |
| Step 11 | (Optional) switch(config)# copy running-config startup-config | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

Configuring Multiprotocol BGP

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

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 | 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 | (Optional) copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config | Saves this configuration change. |

Example

Configuring BMP

You can configure BMP on the Cisco Nexus® 3550-T device.

Before you begin

You must enable BGP (see the Enabling BGP section).

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. |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 3 | bmp server <i>server-number</i> Example: <pre>switch(config-router)# bmp server 1</pre> | <p>Configures the BMP server to which BGP should send information. The server number is used as a key.</p> <p>Note You can configure up to two BMP servers.</p> |
| 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 address of the host and the port number on which the BMP speaker connects to the BMP server. |
| Step 5 | description <i>string</i> 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 { <i>skip</i> / <i>delay time</i> } Example: <pre>switch(config-router)# initial-refresh delay 100</pre> | <p>Configures the option to send a route refresh when BGP is converged and the BMP server connection is established later.</p> <p>The skip option specifies to not send a route refresh if the BMP server connection comes up later.</p> <p>The delay option specifies the time in seconds after which the route refresh should be sent. The range is from 30 to 720 seconds, and the default value is 30 seconds.</p> |
| Step 7 | initial-delay <i>time</i> Example: <pre>switch(config-router)# initial-delay 120</pre> | Configures the delay after which a connection is attempted to the BMP server. The range is from 30 to 720 seconds, and the default value is 45 seconds. |
| Step 8 | stats-reporting-period <i>time</i> Example: <pre>switch(config-router)# stats-reporting-period 50</pre> | Configures the time interval in which the BMP server receives the statistics report from BGP neighbors. The range is from 30 to 720 seconds, and the default is disabled. |
| Step 9 | shutdown Example: <pre>switch(config-router)# shutdown</pre> | Disables the connection to the BMP server. |
| Step 10 | neighbor ip-address Example: | Enters neighbor configuration mode for BGP routing and configures the neighbor IP address. |

| | Command or Action | Purpose |
|----------------|--|---|
| | <pre>switch(config-router)# neighbor 192.168.1.2 switch(config-router-neighbor)#</pre> | |
| Step 11 | remote-as <i>as-number</i> Example: <pre>switch(config-router-neighbor)# remote-as 65535</pre> | Configures the AS number for a remote BGP peer. |
| Step 12 | bmp-activate-server <i>server-number</i> Example: <pre>switch(config-router-neighbor)# bmp-activate-server 1</pre> | Configures the BMP server to which a neighbor's information should be sent. |
| Step 13 | (Optional) show bgp bmp server [<i>server-number</i>] [<i>detail</i>] Example: <pre>switch(config-router-neighbor)# show bgp bmp server</pre> | Displays BMP server information. |
| Step 14 | (Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor)# copy running-config startup-config</pre> | Saves this configuration change. |

About BGP Graceful Shutdown

BGP supports the graceful shutdown feature. This BGP feature works with the BGP **shutdown** command to:

- Dramatically decrease the network convergence time when a router or link is taken offline.
- Reduce or eliminate dropped packets that are in transit when a router or link is taken offline.

Despite the name, BGP graceful shutdown does not actually cause a shutdown. Instead, it alerts connected routers that a router or link will be going down soon.

The graceful shutdown feature uses the GRACEFUL_SHUTDOWN well-known community (0xFFFF0000 or 65535:0), which is identified by IANA and the IETF through RFC 8326. This well-known community can be attached to any routes, and it is processed like any other attribute of a route.

Because this feature announces that a router or link will be going down, the feature is useful in preparation of maintenance windows or planned outages. Use this feature before shutting down BGP to limit the impact on traffic.

Graceful Shutdown Aware and Activate

BGP routers can control the preference of all routes with the GRACEFUL_SHUTDOWN community through the concept of GRACEFUL SHUTDOWN awareness. Graceful shutdown awareness is enabled by default, which enables the receiving peers to deprefer incoming routes carrying the GRACEFUL_SHUTDOWN community. Although not a typical use case, you can disable and reenable graceful shutdown awareness through the **graceful-shutdown aware** command.

Graceful shutdown aware is applicable only at the BGP global context. For information about contexts, see [Graceful Shutdown Contexts, on page 55](#). The aware option operates with another option, the **activate** option, which you can assign to a route map for more granular control over graceful shutdown routes.

Interaction of the Graceful Shutdown Aware and Activate Options

When a graceful shutdown is activated, the GRACEFUL_SHUTDOWN community is appended to route updates only when you specify the **activate** keyword. At this point, new route updates that contain the community are generated and transmitted. When the **graceful-shutdown aware** command is configured, all routers that receive the community then deprefer (lower the route preference of) the routes in the update. Without the **graceful-shutdown aware** command, BGP does not deprefer routes with the GRACEFUL_SHUTDOWN community.

After the feature is activated and the routers are aware of graceful shutdown, BGP still considers the routes with the GRACEFUL_SHUTDOWN community as valid. However, those routes are given the lowest priority in the best-path calculation. If alternate paths are available, new best paths are chosen, and convergence occurs to accommodate the router or link that will soon go down.

Graceful Shutdown Contexts

BGP graceful shutdown feature has two contexts that determine what the feature affects and what functionality is available.

| Context | Affects | Commands |
|---------|--|---|
| Global | The entire switch and all routes processed by it. For example, readvertise all routes with the GRACEFUL_SHUTDOWN community. | graceful-shutdown activate [route-map route-map] graceful-shutdown aware |
| Peer | A BGP peer or a link between neighbors. For example, advertise only one link between peers with GRACEFUL_SHUTDOWN community. | graceful-shutdown activate [route-map route-map] |

Graceful Shutdown with Route Maps

Graceful shutdown works with the route policy manager (RPM) feature to control how the switch's BGP router transmits and receives routes with the `GRACEFUL_SHUTDOWN` community. Route maps can process route updates with the community in the inbound and outbound directions. Typically, route maps are not required. However, if needed, you can use them to customize the control of graceful shutdown routes.

Normal Inbound Route Maps

Normal inbound route maps affect routes that are incoming to the BGP router. Normal inbound route maps are not commonly used with the graceful shutdown feature because routers are aware of graceful shutdown by default.

Cisco Nexus® switches do not require an inbound route map for the graceful shutdown feature. Cisco NX-OS switches have implicit inbound route maps that automatically deprefer any routes that have the `GRACEFUL_SHUTDOWN` community if the BGP router is graceful shutdown aware.

Normal inbound route maps can be configured to match against the well-known `GRACEFUL_SHUTDOWN` community. Although these inbound route maps are not common, there are some cases where they are used:

- If switches are running a Cisco NX-OS release that do not have the implicit inbound route map, a graceful shutdown inbound route map to use the graceful shutdown feature on these switches. The route map must match inbound routes with the well-known `GRACEFUL_SHUTDOWN` community, permit them, and deprefer them. If an inbound route map is needed, create it on the BGP peer that is running a compatible version of NX-OS and is receiving the graceful shutdown routes.
- If you want to disable graceful shutdown aware, but still want the router to act on incoming routes with `GRACEFUL_SHUTDOWN` community from some BGP neighbors, you can configure an inbound route map under the respective peers.

Normal Outbound Route Maps

Normal outbound route maps control forwarding the routes that a BGP router sends. Normal outbound route maps can affect the graceful shutdown feature. For example, you can configure an outbound route map to match on the `GRACEFUL_SHUTDOWN` community and set attributes, and it takes precedence over any graceful shutdown outbound route maps.

Graceful Shutdown Outbound Route Maps

Outbound Graceful shutdown route maps are specific type of outbound route map for the graceful shutdown feature. They are optional, but they are useful when you already have a community list that is associated with a route map. The typical graceful shutdown outbound route map contains only `set` clauses to set or modify certain attributes.

You can use outbound route maps in the following ways:

- For customers that already have existing outbound route maps, you can add a new entry with a higher sequence number, match on the `GRACEFUL_SHUTDOWN` well-known community, and add any attributes that you want.
- You can also use a graceful shutdown outbound route map with the **`graceful-shutdown activate route-map name`** option. This is the typical use case.

This route map requires no match clauses, so the route map matches on all routes being sent to the neighbor.

Route Map Precedence

When multiple route maps are present on the same router, the following order of precedence is applied to determine how routes with the community are processed: Consider the following example. Assume you have a standard outbound route map name Red that sets a local-preference of 60. Also, assume you have a peer graceful-shutdown route map that is named Blue that sets local-pref to 30. When the route update is processed, the local preference will be set to 60 because Red overwrites Blue.

- Normal outbound route maps take precedence over peer graceful shutdown maps.
- Peer graceful shutdown maps take precedence over global graceful shutdown maps.

Guidelines and Limitations

The following are limitations and guidelines for BGP global shutdown:

- Graceful shutdown feature can only help avoid traffic loss when alternative routes exist in the network for the affected routers. If the router has no alternate routes, routes carrying the GRACEFUL_SHUTDOWN community are the only ones available, and therefore, are used in the best-path calculation. This situation defeats the purpose of the feature.
- Configuring a BGP send community is required to send the GRACEFUL_SHUTDOWN community.
- For route maps:
 - When global route maps and neighbor route maps are configured, the per-neighbor route maps take precedence.
 - Outbound route maps take precedence over any global route maps configured for graceful shutdown.
 - Outbound route maps take precedence over any peer route maps configured for graceful shutdown.
 - To add the graceful shutdown functionality to legacy (existing) inbound route maps, follow this order:
 1. Add the graceful shutdown match clause to the top of the route map by setting a low sequence number for the clause (for example, sequence number 0).
 2. Add a continue statement after the graceful shutdown clause. If you omit the continue statement, route-map processing stops when it matches the graceful shutdown clause, any other clauses with higher sequence numbers (for example, 1 and higher) are not processed.

Graceful Shutdown Task Overview

To use the graceful shutdown feature, you typically enable graceful-shutdown aware on all Cisco Nexus switches and leave the feature enabled. When a BGP router must be taken offline, you configure graceful-shutdown activate on it.

The following details document the best practice for using the graceful shutdown feature.

To bring the router or link down:

1. Configure the Graceful Shutdown feature.
2. Watch the neighbor for the best path.
3. When the best path is recalculated, issue the **shutdown** command to disable BGP.
4. Perform the work that required you to shut down the router or link.

To bring the router or link back online:

1. When you finish the work that required the shutdown, reenables BGP (**no shutdown**).
2. Disable the graceful shutdown feature (**no graceful-shutdown activate** in config router mode).

Configuring Graceful Shutdown on a Link

This task enables you to configure graceful shutdown on a specific link between two BGP routers.

Before you begin

If you have not already enabled BGP, enable it now (**feature bgp**).

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | config terminal Example: switch-1# configure terminal switch-1(config)# | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: switch-1(config)# router bgp 110 switch-1(config-router)# | Enters router configuration mode to create or configure a BGP routing process. |
| Step 3 | neighbor { ipv4-address} remote-as as-number Example: switch-1(config-router)# neighbor 10.0.0.3 remote-as 200 switch-1(config-router-neighbor)# | Configures the autonomous system (AS) to which the neighbor belongs. |
| Step 4 | graceful-shutdown activate [route-map map-name] Example: switch-1(config-router-neighbor)# graceful-shutdown activate route-map | Configures graceful shutdown on the link to the neighbor. Also, advertises the routes with the well-known GRACEFUL_SHUTDOWN community and applies the route map to the outbound route updates. |

| | Command or Action | Purpose |
|--|--|--|
| | gshutPeer out switch-1(config-router-neighbor) # | The routes are advertised with the graceful-shutdown community by default. In this example, routes are advertised to the neighbor with the Graceful-shutdown community with a route-map named gshutPeer. The devices receiving the gshut community look at the communities of the route and optionally use the communities to apply routing policy. |

Filtering BGP Routes and Setting Local Preference Based On GRACEFUL_SHUTDOWN Communities

Switches do not have an inbound route map that matches against the GRACEFUL_SHUTDOWN community name. Therefore, they have no way of identifying and depreffering the correct routes.

For switches running a release of NX-OS, you must configure an inbound route map that matches on the community value for graceful shutdown (65535:0) and depreffers routes.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal Example: switch-1# configure terminal switch-1(config)# | Enters global configuration mode. |
| Step 2 | ip community list standard <i>community-list-name seq sequence-number { permit deny } value</i> Example: switch-1(config)# ip community-list standard GSHUT seq 10 permit 65535:0 switch-1(config)# | Configures a community list and permits or denies routes that have the well-known graceful shutdown community value. |
| Step 3 | route map map-tag {deny permit} <i>sequence-number</i> Example: switch-1(config)# route-map RM_GSHUT permit 10 switch-1(config-route-map)# | Configures a route map as sequence 10 and permits routes that have the GRACEFUL_SHUTDOWN community. |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 4 | match community <i>community-list-name</i> Example: <pre>switch-1(config-route-map)# match community GSHUT switch-1(config-route-map)#</pre> | Configures that routes that match the IP community list GSHUT are processed by Route Policy Manager (RPM). |
| Step 5 | set local-preference <i>local-pref-value</i> Example: <pre>switch-1(config-route-map)# set local-preference 10 switch-1(config-route-map)#</pre> | Configures that the routes that match the IP community list GSHUT will be given a specified local preference. |
| Step 6 | exit Example: <pre>switch-1(config-route-map)# exit switch-1(config)#</pre> | Leaves route map configuration and returns to global configuration mode. |
| Step 7 | router bgp <i>community-list-name</i> Example: <pre>switch-1(config)# router bgp 100 switch-1(config-router)#</pre> | Enters router configuration mode and creates a BGP instance. |
| Step 8 | neighbor { ipv4-address } Example: <pre>switch-1(config-router)# neighbor 10.0.0.3 switch-1(config-router-neighbor)#</pre> | Enters route BGP neighbor mode for a specified neighbor. |
| Step 9 | address-family { address-family sub family } Example: <pre>nxosv2(config-router-neighbor)# address-family ipv4 unicast nxosv2(config-router-neighbor-af)#</pre> | Puts the neighbor into address family (AF) configuration mode. |
| Step 10 | send community Example: <pre>nxosv2(config-router-neighbor-af)# send-community nxosv2(config-router-neighbor-af)#</pre> | Enables BGP community exchange with the neighbor. |
| Step 11 | route map <i>map-tag in</i> Example: <pre>nxosv2(config-router-neighbor-af)# route-map RM_GSHUT in nxosv2(config-router-neighbor-af)#</pre> | Applies the route map to incoming routes from the neighbor. In this example, the route map that is named RM_GSHUT permits routes with the GRACEFUL_SHUTDOWN community from the neighbor. |

Configuring Graceful Shutdown for All BGP Neighbors

You can manually apply the GRACEFUL_SHUTDOWN well-known community to all the neighbors of a graceful shutdown initiator.

You can configure graceful shutdown at the global level for all BGP neighbors.

Before you begin

If you have not already enabled BGP, enable it now (**feature bgp**).

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: <pre>switch-1# configure terminal switch-1(config)#</pre> | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system-number</i> Example: <pre>switch-1(config)# router bgp 110 switch-1(config-router)#</pre> | Enters router configuration mode to create or configure a BGP routing process. |
| Step 3 | graceful-shutdown activate [route-map <i>map-name</i>] Example: <pre>switch-1(config-router-neighbor)# graceful-shutdown activate route-map gshutPeer switch-1(config-router-neighbor)#</pre> | <p>Configures graceful shutdown route map for the links to all neighbors. Also, advertises all routes with the well-known GRACEFUL_SHUTDOWN community and applies the route map to the outbound route updates.</p> <p>The routes are advertised with the GRACEFUL_SHUTDOWN community by default. In this example, routes are advertised to all neighbors with the community with a route-map named gshutPeer. The route map should contain only set clauses.</p> <p>The devices receiving the GRACEFUL_SHUTDOWN community look at the communities of the route and optionally use the communities to apply routing policy.</p> |

Controlling the Preference for All Routes with the GRACEFUL_SHUTDOWN Community

Cisco NX-OS enables lowering the preference of incoming routes that have the GRACEFUL_SHUTDOWN community. When **graceful shutdown aware** is enabled, BGP considers routes carrying the community as the lowest preference during best path calculation. By default, lowering the preference is enabled, but you can selectively disable this option.

Whenever you enable or disable this option, you trigger a BGP best-path calculation. This option gives you the flexibility to control the behavior of the BGP best-path calculation for the graceful shutdown well-known community.

Before you begin

If you have not enabled BGP, enable it now (**feature bgp**).

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal Example: <pre>switch-1(config)# config terminal switch-1(config)#</pre> | Enters global configuration mode. |
| Step 2 | router bgp <i>autonomous-system</i> Example: <pre>switch-1(config)# router bgp 100 switch-1(config-router)#</pre> | Enters router configuration mode and configures a BGP routing process. |
| Step 3 | (Optional) no graceful-shutdown aware Example: <pre>switch-1(config-router)# no graceful-shutdown aware switch-1(config-router)#</pre> | For this BGP router, do not give lower preference for all routes that have the GRACEFUL_SHUTDOWN community. The default action is to deprefer routes when the graceful shutdown aware feature is disabled, so using the no form of the command is optional for not deprefering graceful shutdown routes. |

Preventing Sending the GRACEFUL_SHUTDOWN Community to a Peer

If you no longer need the GRACEFUL_SHUTDOWN community that is appended as a route attribute to outbound route updates, you can remove the community, which no longer sends it to a specified neighbor. One use case would be when a router is at an autonomous system boundary, and you do not want the graceful shutdown functionality to propagate outside of an autonomous system boundary.

To prevent sending the GRACEFUL_SHUTDOWN to a peer, you can disable the send community option or strip the community from the outbound route map.

Choose either of the following methods:

- Disable the send-community in the running config.

Example:

```
nxosv2(config-router-neighbor-af) # no send-community standard
nxosv2(config-router-neighbor-af) #
```

If you use this option, the GRACEFUL_SHUTDOWN community is still received by the switch, but it is not sent to the downstream neighbor through the outbound route map. All standard communities are not sent either.

- Delete the GRACEFUL_SHUTDOWN community through an outbound route map by following these steps:
 1. Create an IP community list matches the GRACEFUL_SHUTDOWN community.
 2. Create an outbound route map to match against the GRACEFUL_SHUTDOWN community.
 3. Use a **set community-list delete** clause to strip GRACEFUL_SHUTDOWN community.

If you use this option, the community list matches and permits the GRACEFUL_SHUTDOWN community, then the outbound route map matches against the community and then deletes it from the outbound route map. All other communities pass through the outbound route map without issue.

Displaying Graceful Shutdown Information

Information about the graceful shutdown feature is available through the following **show** commands.

| Command | Action |
|---|---|
| show ip bgp community-list graceful-shutdown | Shows all entries in the BGP routing table that have the GRACEFUL_SHUTDOWN community. |
| show running-config bgp | Shows the running BGP configuration. |
| show running-config bgp all | Shows all information for the running BGP configuration including information about the graceful shutdown feature. |
| show bgp address-family neighbors neighbor-address Note In Cisco Nexus 3550-T BGP only supports IPv4 unicast address family. | When the feature is configured for the peer, shows the following: <ul style="list-style-type: none"> • The state of the graceful-shutdown-activate feature for the specified neighbor • The name of any graceful shutdown route map configured for the specified neighbor |

| Command | Action |
|----------------------------|---|
| show bgp process | <p>Shows different information depending on the context.</p> <p>When the graceful-shutdown-activate option is configured in peer context, shows the enabled or disabled state for the feature through <code>graceful-shutdown-active</code>.</p> <p>When the graceful-shutdown-activate option is configured in global context and has a graceful-shutdown route map, shows the enabled state of the feature through the following:</p> <ul style="list-style-type: none"> • <code>graceful-shutdown-active</code> • <code>graceful-shutdown-aware</code> • <code>graceful-shutdown route-map</code> |
| show ip bgp address | <p>For the specified address, shows the BGP routing table information, including the following:</p> <ul style="list-style-type: none"> • The state of the specified address as the best path • Whether the specified address is part of the GRACEFUL_SHUTDOWN community |

Graceful Shutdown Configuration Examples

These examples show some configurations for using the graceful shutdown feature.

Configuring Graceful Shutdown for a BGP Link

The following example shows how to configure graceful shutdown while setting a local preference and a community:

- Configuring graceful shutdown activate for the link to the specified neighbor
- Adding the GRACEFUL_SHUTDOWN community to the routes
- Setting a route map named `gshutPeer` with only set clauses for outbound routes with the community.

```
router bgp 100
  neighbor 20.0.0.3 remote-as 200
    graceful-shutdown activate route-map gshutPeer
    address-family ipv4 unicast
      send-community

route-map gshutPeer permit 10
  set local-preference 0
  set community 200:30
```


Configuring Graceful Shutdown for All-Neighbor BGP Links

The following example shows:

- Configuring graceful shutdown activate for all the links connecting the local router and all its neighbors.
- Adding the GRACEFUL_SHUTDOWN community to the routes.
- Setting a route map that is named gshutAall with only set clauses for all outbound routes.

```
router bgp 200
  graceful-shutdown activate route-map gshutAll

route-map gshutAll permit 10
  set as-path prepend 10 100 110
  set community 100:80

route-map Red permit 10
  set local-pref 20

router bgp 100
  graceful-shutdown activate route-map gshutAll
  router-id 2.2.2.2
  address-family ipv4 unicast
  network 2.2.2.2/32
  neighbor 1.1.1.1 remote-as 100
  update-source loopback0
  address-family ipv4 unicast
  send-community
  neighbor 20.0.0.3 remote-as 200
  address-family ipv4 unicast
  send-community
  route-map Red out
```

In this example, the `gshutAll` route-map takes effect for neighbor 1.1.1.1, but not neighbor 20.0.0.3, because the outbound route-map `Red` configured under neighbor 20.0.0.3 takes precedence instead.

Configuring Graceful Shutdown Under a Peer-Template

This example configures the graceful shutdown feature under a peer-session template, which is inherited by a neighbor.

```
router bgp 200
  template peer-session pl
    graceful-shutdown activate route-map gshut_out
  neighbor 1.1.1.1 remote-as 100
  inherit peer-session pl
  address-family ipv4 unicast
  send-community
```

Filtering BGP Routes and Setting Local Preference Based on GRACEFUL_SHUTDOWN Community Using and Inbound Route Map

This example shows how to use a community list to filter the incoming routes that have the GRACEFUL_SHUTDOWN community. This configuration is useful for legacy switches that are not running Cisco NX-OS 9.3(1) as a minimum version.

The following example shows:

- An IP Community List that permits routes that have the GRACEFUL_SHUTDOWN community.

- A route map that is named RM_GSHUT that permits routes based on a standard community list named GSHUT.
- The route map also sets the preference for the routes it processes to 0 so that those routes are given lower preference for best path calculation when the router goes offline. The route map is applied to incoming IPv4 routes from the neighbor (20.0.0.2).

```
ip community-list standard GSHUT permit 65535:0

route-map RM_GSHUT permit 10
  match community GSHUT
  set local-preference 0

router bgp 200
  neighbor 20.0.0.2 remote-as 100
  address-family ipv4 unicast
    send-community
  route-map RM_GSHUT in
```

Configuring a Graceful Restart

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

Before you begin

You must enable BGP (see the Enabling BGP section).

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 | (Optional) timers prefix-peer-timeout <i>timeout</i> Example: <pre>switch(config-router)# timers prefix-peer-timeout 20</pre> | Configures the timeout value (in seconds) for BGP prefix peers. The default value is 90 seconds. |
| Step 4 | graceful-restart Example: <pre>switch(config-router)# graceful-restart</pre> | Enables a graceful restart and the graceful restart helper functionality. This command is enabled by default. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | This command triggers an automatic notification and session reset for the BGP neighbor sessions. |
| Step 5 | graceful-restart {restart-time time stalepath-time time} 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 6 | 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> |
| Step 7 | (Optional) show running-config bgp Example: <pre>switch(config-router)# show running-config bgp</pre> | Displays the BGP configuration. |
| Step 8 | (Optional) copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre> | 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)# graceful-restart restart-time 300
switch(config-router)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

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



Note *Cisco Nexus 3550-T - 10.1(2t) release*, supports only default VRF and management VRF.

| Command | Purpose |
|--|---|
| <code>show bgp all [summary]</code> | Displays the BGP information for all address families. |
| <code>show bgp convergence</code> | Displays the BGP information for all address families. |
| <code>show bgp {ipv4} {unicast} [ip-address] community {regex expression [community] [no-advertise] [no-export] [no-export-subconfed]}</code> | Displays the BGP routes that match a BGP community. |
| <code>show bgp {ipv4} {unicast} [ip-address] community-list list-name</code> | Displays the BGP routes that match a BGP community list. |
| <code>show bgp {ipv4} {unicast} [ip-address] extcommunity {regex expression generic [non-transitive transitive] aa4:nn [exact-match]}</code> | Displays the BGP routes that match a BGP extended community. |
| <code>show bgp {ipv4} {unicast} [ip-address] extcommunity-list list-name [exact-match]</code> | Displays the BGP routes that match a BGP extended community list. |
| <code>show bgp {ipv4} {unicast} [ip-address] extcommunity-list list-name [exact-match]</code> | Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information. |
| <code>show bgp {ipv4} {unicast} [ip-address] {dampening dampened-paths [regex expression]}</code> | Displays the BGP route history paths. |
| <code>show bgp {ipv4 vpnv4} {unicast} [ip-address] filter-list list-name</code> | Displays the information for the BGP filter list. |
| <code>show bgp {ipv4 vpnv4} {unicast} [ip-address] neighbors [ip-address]</code> | Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors. |
| <code>show bgp {ipv4} {unicast} [ip-address] {nexthop nexthop-database}</code> | Displays the information for the BGP route next hop. |
| <code>show bgp paths</code> | Displays the BGP path information. |

| Command | Purpose |
|--|--|
| show bgp {ipv4} {unicast} [ip-address] policy name | Displays the BGP policy information. Use the clear bgp policy command to clear the policy information. |
| show bgp {ipv4} {unicast} [ip-address] prefix-list list-name | Displays the BGP routes that match the prefix list. |
| show bgp {ipv4} {unicast} [ip-address] received-paths | Displays the BGP paths stored for soft reconfiguration. |
| show bgp {ipv4} {unicast} [ip-address] regex expression | Displays the BGP routes that match the AS_path regular expression. |
| show bgp {ipv4} {unicast} [ip-address] route-map map-name | Displays the BGP routes that match the route map. |
| show bgp peer-policy name | Displays the information about BGP peer policies. |
| show bgp peer-session name | Displays the information about BGP peer sessions. |
| show bgp peer-template name | Displays the information about BGP peer templates. Use the clear bgp peer-template command to clear all neighbors in a peer template. |
| show bgp process | Displays the BGP process information. |
| show ip route ip-address detail vrf all i bw | 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. |
| show {ipv4} bgp options | Displays the BGP status and configuration information. |
| show running-configuration bgp | Displays the current running BGP configuration. |

Monitoring BGP Statistics

To display BGP statistics, use the following commands:

| Command | Purpose |
|---------|---------|
|---------|---------|

| | |
|----------------------------|---|
| | Displays the BGP route flap statistics. Use the clear bgp flap-statistics command to clear these statistics. |
| | Displays injected routes in the routing table. |
| show bgp sessions | Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics. |
| show bgp statistics | Displays the BGP statistics. |

Related Topics

The following topics can give more information on BGP:

- [Configuring Basic BGP](#)

Additional References

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