Configuring a BGP Route Server

BGP route server is a feature designed for internet exchange (IX) operators that provides an alternative to full eBGP mesh peering among the service providers who have a presence at the IX. The route server provides eBGP route reflection with customized policy support for each service provider. That is, a route server context can override the normal BGP best path for a prefix with a different path based on a policy, or suppress all paths for a prefix and not advertise the prefix. The BGP route server provides reduced configuration complexity and reduced CPU and memory requirements on each border router. The route server also reduces overhead expense incurred by individualized peering agreements.

- Finding Feature Information, page 1
- Information About BGP Route Server, page 2
- How to Configure a BGP Route Server, page 8
- Configuration Examples for BGP Route Server, page 16
- Additional References, page 19
- Feature Information for BGP Route Server, page 20

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About BGP Route Server

The Problem That a BGP Route Server Solves

In order to understand the problem that a BGP route server solves, it is helpful to understand the following information about service provider (SP) peering and the eBGP mesh that results from public peering.

Private vs. Public Peering of Service Providers

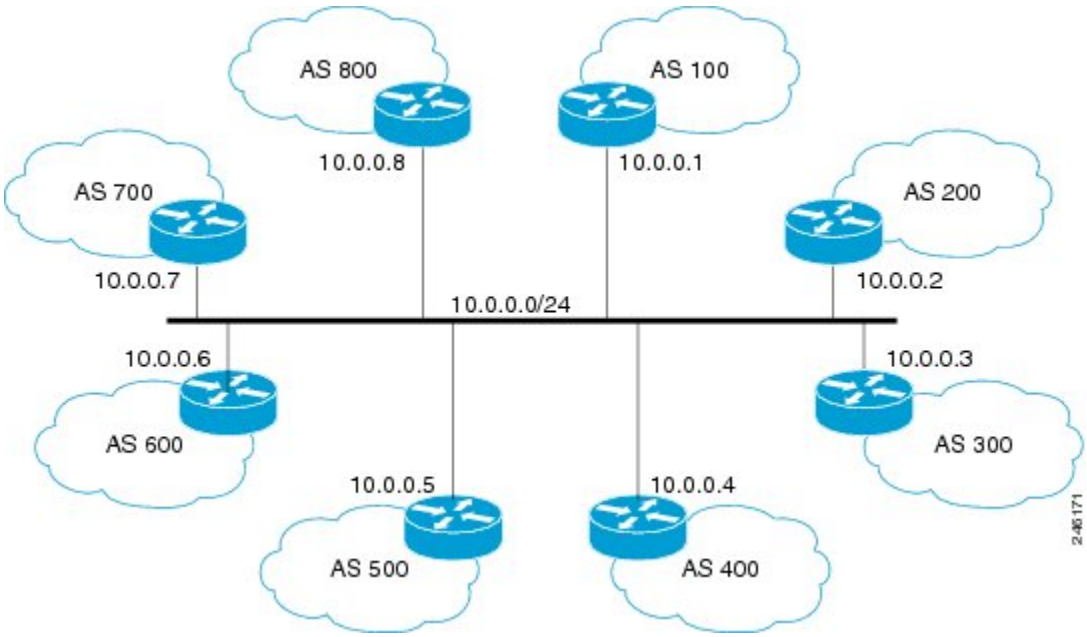
Peering is the connecting of two service providers (SPs) for the purpose of exchanging network traffic between them. Peerings are either private or public.

- In a private peering, two SPs that want to connect decide on a physical site where their networks can be connected and negotiate a contract that covers the details of the connection arrangement. The two parties provide all of the physical space, network equipment, and services (such as electricity and cooling) required to operate the peering connections.
- A public internet exchange (IX), also called a network access point (NAP), is a physical location operated to facilitate the interconnection of multiple SP networks using a shared infrastructure. The IX provides the physical necessities such as rack space for networking devices, electricity, cooling, and a common switching infrastructure required for SPs to directly connect their networks. Unlike private peering, which is typically one-to-one, the IX allows an SP that has a presence at the exchange to connect to multiple peers at a single physical location. The IX provides an alternative to private peering for smaller SPs who do not have the resources required to maintain numerous private peering connections.

Public Peering of SPs within an IX Using BGP

Within the IX, each SP maintains a BGP border router connected to the common switching infrastructure or subnet, as shown in the figure below. In this example, eight different SPs with AS numbers 100 through 800 are connected to the 10.0.0.0/24 subnet through their BGP border routers addressed 10.0.0.1 through 10.0.0.8.

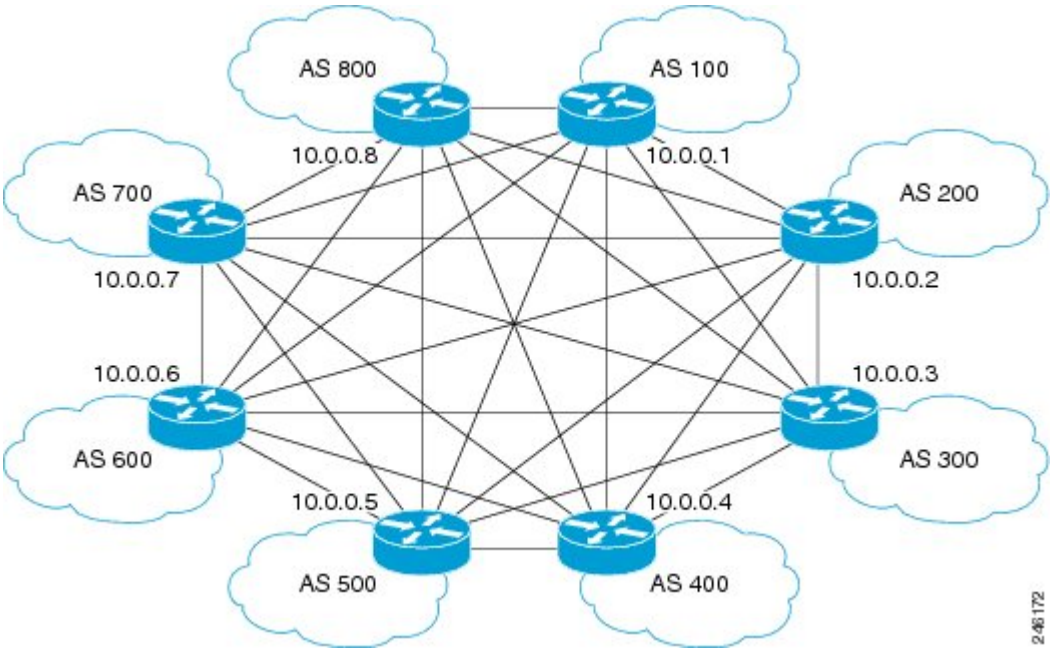
Figure 1: IX Shared Switching Infrastructure



Although each SP's border router is attached to the shared subnet, BGP sessions between each of the SPs must still be configured and maintained individually, for every other SP with which a given SP wants to establish a peering relationship.

Assuming that each SP wants to connect to every other SP, the resulting full mesh of BGP sessions established is shown in the figure below.

Figure 2: IX eBGP Full Mesh



Just as the required iBGP full mesh in an autonomous system presents a scaling and administrative challenge within an SP network, the eBGP full mesh required for peering at an IX presents a challenge for eBGP, for these reasons:

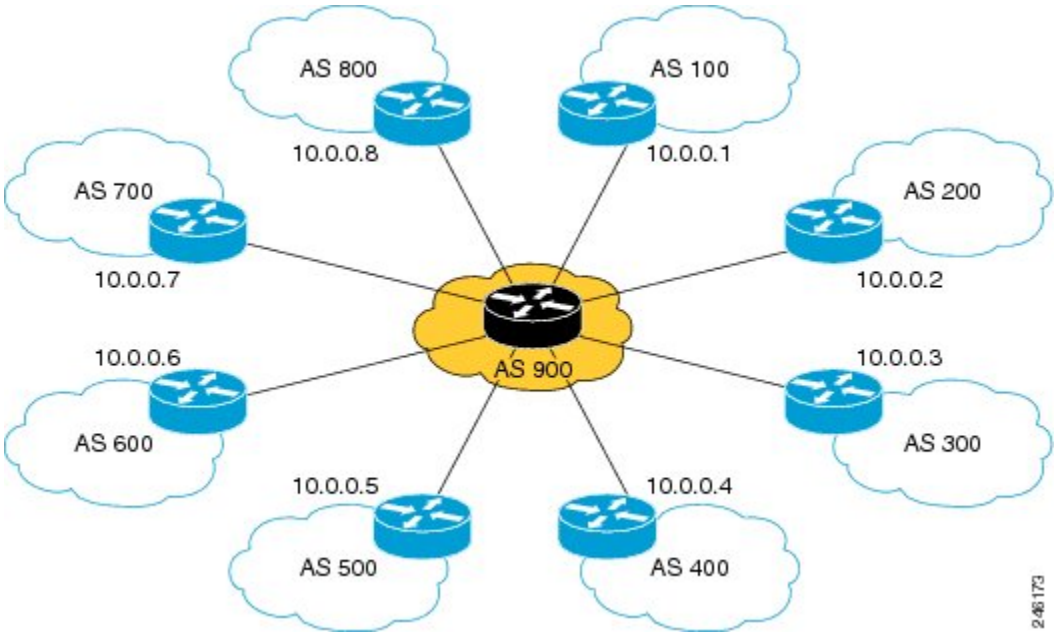
- The full mesh of direct peering sessions requires a BGP session to be configured and maintained for each connection.
- There is additional operational overhead from contracts that would need to be negotiated with each SP peer connecting to a given provider at the IX.

Because larger global SPs might have a presence at dozens or hundreds of internet exchanges worldwide, and dozens or hundreds of potential peers at each IX, it would be a huge operational expense to connect to all of the small providers. Consequently, the state of peering prior to the BGP Route Server feature is that a large global SP connects to only a subset of other large providers to limit the management and operational overhead. A more scalable alternative to direct peering would allow large global SPs to connect to more small providers.

BGP Route Server Simplifies SP Interconnections

A BGP route server simplifies interconnection of SPs at an IX, as shown in the figure below.

Figure 3: IX with eBGP Route Server

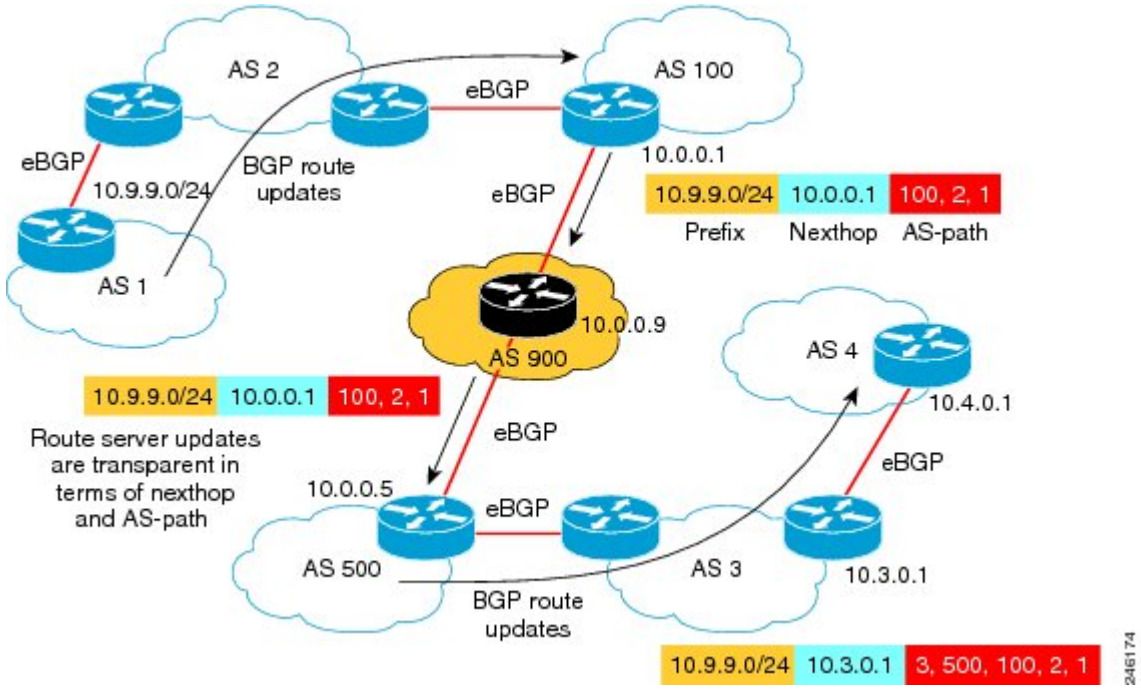


Instead of maintaining individual, direct eBGP peerings with every other provider, an SP maintains only a single connection to the route server operated by the IX. Peering with only the route server reduces the configuration complexity on each border router, reduces CPU and memory requirements on the border routers, and avoids most of the operational overhead incurred by individualized peering agreements.

The route server provides AS-path, MED, and nexthop transparency so that peering SPs at the IX still appear to be directly connected. In reality, the IX route server mediates this peering, but that relationship is invisible outside of the IX.

The figure below illustrates an example of transparent route propagation with a route server at an IX.

Figure 4: Transparent Route Propagation with Route Server at IX



In the figure above, a routing update goes from AS 1 to AS 2 to AS 100. The update leaves the router in AS 100 advertising that the router can reach the prefix 10.9.9.0/24, use 10.0.0.1 as the next hop, and use the AS path of AS100, AS2, AS1.

The router in AS 900 is a route server and the router in AS 500 is a route server client. A route server client receives updates from a route server. As shown in the figure above, the router in AS 900 does not change the update; route server updates are transparent in terms of MED, next hop and AS-path. The update goes to the client with the same prefix, next hop and AS-path that came from the router at 10.0.0.1.

Benefits of a BGP Route Server

A BGP route server provides the following benefits:

- Reduced configuration complexity on each border router.
- Reduced CPU and memory requirements on each border router.
- Reduced operational overhead incurred by individualized peering agreements.
- The ability for a route server context to override the normal BGP best path with an alternative path based on some policy.
- The ability for a route server context to suppress all paths for a prefix and therefore not advertise the prefix.

Route Server Context Provides Flexible Routing Policy

A BGP route server can provide a flexible routing policy. Your network environment might or might not have routes that need a customized (flexible) policy handling.

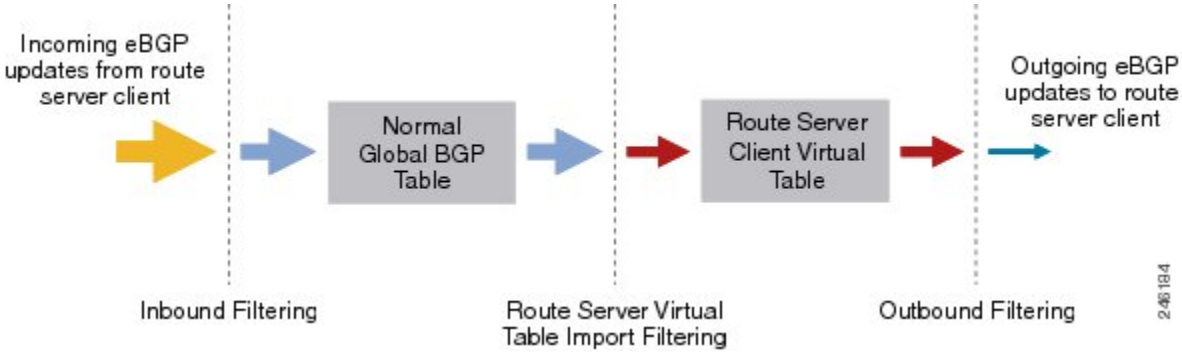
Routes needing flexible policy handling are selected for import into a route server context by configuring an import map. The import map references a route map, where the actual policy is defined by at least one **permit** statement. Routes (paths) that match the route map are included in a second "best path" calculation. The resulting best path, if it is different from the global best path, is imported into the context. Router server clients associated with a router server context will override the global best path with the context's best path when sending routing updates.

Multiple contexts can be created, and the appropriate context is referenced on the route server by the neighbors assigned to use that context (in the **neighbor route-server-client** command). Thus, multiple neighbors sharing the same policy can share the same route server context.

Three Stages of Filtering on a Route Server Client

With the introduction of route server context, there are now three stages of route filtering that can be applied to a route server client, as shown in the figure below. The three stages of filtering are described below the figure. You can apply all, none, or any combination of the three filtering methods to a route server client. In the figure, the decreasing arrow sizes symbolize that potentially fewer routes might pass each filter than entered the filter.

Figure 5: Route Server Filtering in Three Stages



1. As shown in the figure above beginning at the left, when incoming eBGP updates arrive from a route server client, the system will apply inbound route filters for a route server client the same way it does for a non-route-server client (configured with the **neighbor route-map in** command). All routes permitted by the client's inbound filtering are installed in the global BGP table for the appropriate address family, as usual, and anything else is dropped.
2. If any route server contexts have been configured with flexible policy using the **import-map** command, the best path from among the subset of matching routes is imported into the virtual table for the contexts. Route server clients associated with a context will then override any routes from the global BGP table with customized routes from the context's virtual table when generating updates.

3 A route server client's outbound filtering policies (configured with the **neighbor route-map out** command) will be applied to the global updates that do not have customized policy, and the outbound filtering policies are also applied to any updates generated from the route server context's virtual table.

How to Configure a BGP Route Server

Configure a Route Server with Basic Functionality

Perform this task to configure a BGP router as route server for IPv4 or IPv6. This task will enable the basic route server functionality for nexthop, AS-path, and MED transparency.

Note This task does not enable flexible policy handling. To enable flexible policy handling, see the Configure a Route Server with Flexible Policy Handling, on page 11.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*
5. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
6. **neighbor** {*ipv4-address*| *ipv6-address*} **activate**
7. **neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 100`	Adds an entry to the BGP neighbor table.
Step 5	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 6	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 7	**neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 route-server-client`	Configures the BGP neighbor at the specified address to be a route server client.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Ends the current configuration and returns to privileged EXEC mode.

Configure a Route Server Client To Receive Updates

In the prior task, you configured a route server. A route server does not put its own AS number in the AS-path; there is AS-path transparency. This means the route server client will receive updates in which the first AS number in the AS-path is not the sending router's AS number.

By default, a router denies an update received from an eBGP peer that does not list its AS number at the beginning of the AS-path in an incoming update. Therefore, you must disable that behavior on the client in order for the client to receive the updates. To do so, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp enforce-first-as**
5. **neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*
6. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
7. **neighbor** {*ipv4-address*| *ipv6-address*} **activate**
8. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**no bgp enforce-first-as** **Example:** `Router(config-router)# no bgp enforce-first-as`	Disables requirement that an update received from an eBGP peer list its AS number at the beginning of the AS_PATH. • By default, a router is configured to deny an update received from an external BGP (eBGP) peer that does not list its autonomous system number at the beginning of the AS_PATH in the incoming update. • In order to receive updates from the route server, which will not have its AS first in the AS_PATH, specify **no bgp enforce-first-as**to disable the enforcement.
Step 5	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*	Adds an entry to the BGP neighbor table.

	Command or Action	Purpose
	Example: `Router(config-router)# neighbor 10.0.0.1 remote-as 100`	
Step 6	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 7	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.

Configure a Route Server with Flexible Policy Handling

Perform this task if you need your BGP route server to support a customized, flexible policy in addition to basic route server functionality.

In order to configure flexible policy handling, create a route server context, which includes an import map. The import map references a standard route map.

In this particular configuration task, the policy is based on autonomous system number, so the **match as-path** command is used. The actual AS number is identified in the **ip as-path access-list** command. You may match on nexthop, AS path, communities, and extended communities.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **route-server-context** *context-name*
5. **description** *string*
6. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
7. **import-map** *route-map-name*
8. **exit-address-family**
9. **exit-route-server-context**
10. **exit**
11. **ip as-path access-list** *access-list-number* {**permit**| **deny**} *regexp*
12. **route-map** *route-map-name* [**permit** | **deny**] *sequence-number*
13. **match as-path** *access-list-number*
14. **exit**
15. **router bgp** *autonomous-system-number*
16. **neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*
17. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
18. **neighbor** {*ipv4-address*| *ipv6-address*} **activate**
19. **neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client context** *ctx-name*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**route-server-context** *context-name*	Creates a route server context.

	Command or Action	Purpose
	Example: `Router(config-router)# route-server-context ONLY_AS27_CONTEXT`	• In this example, a context named ONLY_AS27_CONTEXT is created.
Step 5	**description** *string* **Example:** `Router(config-router-rsctx)# description Permit only routes with AS 27 in AS path.`	(Optional) Allows you to describe the context. • Up to 80 characters are allowed.
Step 6	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router-rsctx)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 7	**import-map** *route-map-name* **Example:** `Router(config-router-rsctx-af)# import-map only_AS27_routemap`	Configures flexible policy handling by using the route map that you will create in Step 12 to control which routes will be added to the route server client virtual table.
Step 8	**exit-address-family** **Example:** `Router(config-router-rsctx-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**exit-route-server-context** **Example:** `Router(config-router-rsctx)# exit-route-server-context`	Exits route server context configuration mode.
Step 10	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode.
Step 11	**ip as-path access-list** *access-list-number* {**permit**| **deny**} *regexp* **Example:** `Router(config)# ip as-path access-list 5 permit 27`	Configures an AS path filter using a regular expression. • The **ip as-path** command is not necessarily the command you have to use. Determine what policy you want to create.

	Command or Action	Purpose
Step 12	**route-map** *route-map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config)# route-map only_AS27_routemap permit 10`	Defines whether AS paths that match the subsequent **match as-path**command will be permitted or denied in the route map. • Use the same *route-map-name* that you specified in the **import-map** command above.
Step 13	**match as-path** *access-list-number* **Example:** `Router(config-route-map)# match as-path 5`	Identifies an access list that determines which AS paths are matched and become part of the route map configured in the prior step. • This particular example references the *access-list-number* configured in the **ip as-path access-list** command. • The **match as-path** command is not necessarily the command you have to use. Determine what policy you want to use. • You may match on nexthop, AS path, communities, and extended communities.
Step 14	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode.
Step 15	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 16	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 500`	Adds an entry to the BGP neighbor table.
Step 17	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.

	Command or Action	Purpose
Step 18	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 19	**neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client context** *ctx-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 route-server-client context ONLY_AS27_CONTEXT`	Configures the BGP neighbor at the specified address to be a route server client. • In this example, the route server client at this specified address is assigned to the context called ONLY_AS27_CONTEXT.
Step 20	**end** **Example:** `Router(config-router-af)# end`	Ends the current configuration and returns to privileged EXEC mode.

Displaying BGP Route Server Information and Troubleshooting Route Server

From privileged EXEC mode, perform either of the steps in this task on a BGP route server to see information about the route server.

On a BGP route server client (not the route server), you can use the **show ip bgp ipv4 unicast** or **show ip bgp ipv6 unicast**command to display routing information.

SUMMARY STEPS

1. **enable**
2. **show ip bgp** {**ipv4** | **ipv6**} **unicast route-server** {**all** | {**context** *context-name*}} [**summary**]
3. **debug ip bgp route-server** {**client** | **context** | **event** | **import** | **policy**} [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip bgp** {**ipv4** | **ipv6**} **unicast route-server** {**all** | {**context** *context-name*}} [**summary**] **Example:** `Router#`	Displays the paths chosen for a particular route server context, which migh include the global bestpath, an overriding policy path, or a suppressed path.
Step 3	**debug ip bgp route-server** {**client** | **context** | **event** | **import** | **policy**} [**detail**] **Example:** `Router# debug ip bgp route-server client`	Turns on debugging for BGP route server. **Caution** The **detail** keyword is used for more complex issues and should only be turned on when debugging with a Cisco representative.

Configuration Examples for BGP Route Server

Example BGP Route Server with Basic Functionality

In the following example, the neighbor at 10.0.0.1 is a route server client.

```
router bgp 65000
 neighbor 10.0.0.1 remote-as 100
 neighbor 10.0.0.5 remote-as 500
 address-family ipv4 unicast
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 route-server-client
!
```

Example BGP Route Server Context for Flexible Policy (IPv4 Addressing)

In the following example, the local router is a BGP route server. Its neighbors at 10.10.10.12 and 10.10.10.13 are its route server clients. A route server context named ONLY_AS27_CONTEXT is created and applied to the neighbor at 10.10.10.13. The context uses an import map that references a route map named only_AS27_routemap. The route map matches routes permitted by access list 27. Access list 27 permits routes that have 27 in the AS path.

```
router bgp 65000
   route-server-context ONLY_AS27_CONTEXT
      address-family ipv4 unicast
         import-map only_AS27_routemap
      exit-address-family
   exit-route-server-context
   !
   neighbor 10.10.10.12 remote-as 12
   neighbor 10.10.10.12 description Peer12
   neighbor 10.10.10.13 remote-as 13
   neighbor 10.10.10.13 description Peer13
   neighbor 10.10.10.21 remote-as 21
   neighbor 10.10.10.27 remote-as 27
```

```
   !
   address-family ipv4
      neighbor 10.10.10.12 activate
      neighbor 10.10.10.12 route-server-client
      neighbor 10.10.10.13 activate
      neighbor 10.10.10.13 route-server-client context ONLY_AS27_CONTEXT
      neighbor 10.10.10.21 activate
      neighbor 10.10.10.27 activate
   exit-address-family
!
ip as-path access-list 27 permit 27
!
route-map only_AS27_routemap permit 10
   match as-path 27
!
```

Example Using Show Commands to See That Route Server Context Routes Overwrite Normal Bestpath

In the following output, a BGP route server has two routes from AS 21 that have been selected as best:

```
Route-Server# show ip bgp ipv4 unicast
BGP table version is 31, local router ID is 100.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.10.10.21             23             0 21 ?
*                   10.10.10.27            878             0 27 89 ?
*  100.1.1.1/32     10.10.10.27            878             0 27 89 ?
*>                  10.10.10.21             23             0 21 ?
```

For Peer12, which has been configured as a route-server client, but not associated with any context, the bestpath is advertised in the following output. Note that AS-path, MED, and nexthop transparency have been maintained; the routes look as if they had not passed through the route server.

```
Peer12# show ip bgp ipv4 unicast

BGP table version is 31, local router ID is 10.10.10.12
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.10.10.21             23             0 21 ?
*> 100.1.1.1/32     10.10.10.21             23             0 21 ?
```

Peer13 has also been configured as a route-server client, and it has been associated with a context named ONLY_AS27_CONTEXT. The context references a route map that permits only routes that contain AS 27 in the AS path. This means that the route-server should not send any routes to Peer13 unless they contains AS 27. In our scenario, the route server indeed sends the routes learned via AS 27, even though the routes learned via AS 21 are marked as best. The output below demonstrates that the normal best path was overriden by the best path based on policy. Again, MED, as-path, and nexthop transparency have been maintained.

```
Peer13# show ip bgp ipv4 unicast

BGP table version is 25, local router ID is 10.10.10.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.10.10.27            878             0 27 89 ?
*> 100.1.1.1/32     10.10.10.27            878             0 27 89 ?
```

Example BGP Route Server Context with No Routes Satisfying the Policy

It is possible that, due to policy, no routes are sent to a client even though paths exist. For instance, if we take the prior example and change ONLY_AS27_CONTEXT to ONLY_AS100_CONTEXT, no paths would satisfy this policy and no routes will be sent to the client. The following is the configuration and resulting show output:

```
Route-Server# show run | begin router bgp
router bgp 1
   route-server-context ONLY_AS100_CONTEXT
     !
     address-family ipv4 unicast
        import-map only_AS100_routemap
     exit-address-family
   exit-route-server-context
   !
   neighbor 10.10.10.13 remote-as 13
   neighbor 10.10.10.13 description Peer13
   neighbor 10.10.10.21 remote-as 21
   neighbor 10.10.10.27 remote-as 27
   !
      address-family ipv4
         neighbor 10.10.10.13 activate
         neighbor 10.10.10.13 route-server-client context ONLY_AS100_CONTEXT
         neighbor 10.10.10.21 activate
         neighbor 10.10.10.27 activate
   exit-address-family

!
ip as-path access-list 100 permit 100
!
!
route-map only_AS100_routemap permit 10
   match as-path 100
!
```

Because no routes satisfy the policy, no routes appear in the table of Peer13:

```
Peer13# show ip bgp ipv4 unicast
```

Example BGP Route Server Context for Flexible Policy (IPv6 Addressing)

In the following example under address-family IPv6, the local router is a BGP route server. Its neighbors at 2001:DB8:1::112 and 2001:DB8:1::113 are its route server clients. A route server context named ONLY_AS27_CONTEXT is created and applied to the neighbor at 2001:DB8:1::113. The context uses an import map that references a route map named only_AS27_routemap. The route map matches routes permitted by access list 27. Access list 27 permits routes that have 27 in the AS path.

```
Route-Server# show run | begin router bgp
router bgp 1
   route-server-context ONLY_AS27_CONTEXT
      address-family ipv6 unicast
         import-map only_AS27_routemap
      exit-address-family
   exit-route-server-context
   !
   neighbor 2001:DB8:1::112 remote-as 12
   neighbor 2001:DB8:1::112 description Peer12
   neighbor 2001:DB8:1::113 remote-as 13
   neighbor 2001:DB8:1::113 description Peer13
 !
   address-family ipv6
      neighbor 2001:DB8:1::112 activate
```

```
      neighbor 2001:DB8:1::112 route-server-client
      neighbor 2001:DB8:1::113 activate
      neighbor 2001:DB8:1::113 route-server-client context ONLY_AS27_CONTEXT
 exit-address-family
!
ip as-path access-list 27 permit 27
!
route-map only_AS27_routemap permit 10
   match as-path 27
!
Route-Server#show ip bgp ipv6 unicast route-server all summary

Route server clients without assigned contexts:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:1::112 4           12      19      19        4    0    0 00:12:50        2
Route server clients assigned to context ONLY_AS27_CONTEXT:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:1::113 4           13      23      22        4    0    0 00:16:23        2
```

For Peer12, which has been configured as a route-server client, but not associated with any context, the bestpath is advertised. Note that AS-path, MED, and nexthop transparency have been maintained; the routes look as if they had not passed through the route server.

```
Peer12# show ip bgp ipv6 unicast
BGP table version is 9, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
* 2001:DB8:1::/64  2001:DB8::113            0             0 13 ?
*>                  ::                      0         32768 ?
* 2001:DB8:2::/64  2001:DB8::113            0             0 13 ?
*>                  ::                      0         32768 ?
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	*Cisco IOS IP Routing: BGP Command Reference*
BGP configuration tasks	*IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S*

MIBs

MIB	MIBs Link
• --	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Route Server

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for BGP Route Server

Feature Name	Releases	Feature Information
BGP Route Server	Cisco IOS XE Release 3.3S 15.2(3)T	BGP route server is a feature designed for internet exchange (IX) operators that provides an alternative to full eBGP mesh peering among the service providers who have a presence at the IX. The route server provides eBGP route reflection with customized policy support for each service provider. That is, a route server context can override the normal BGP best path for a prefix with a different path based on a policy, or suppress all paths for a prefix and not advertise the prefix. The BGP route server provides reduced configuration complexity and reduced CPU and memory requirements on each border router. The route server also reduces overhead expense incurred by individualized peering agreements. The following commands were introduced: • **debug ip bgp route-server** • **description** (route server context) • **exit-route-server-context** • **import-map** • **neighbor route-server-client** • **route-server-context** • **show ip bgp unicast route-server**