BGP Route-Map Continue

The BGP Route-Map Continue feature introduces the continue clause to BGP route-map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configuration need not be repeated within the same route map.

- Finding Feature Information, on page 1
- Information About BGP Route Map Continue, on page 1
- How to Filter Traffic Using Continue Clauses in a BGP Route Map, on page 3
- Configuration Examples for BGP Route Map Continue, on page 6
- Additional References, on page 8
- Feature Information for BGP Route Map Continue, on page 8

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 Map Continue

BGP Route Map with a Continue Clause

In BGP route-map configuration, the continue clause allows for more programmable policy configuration and route filtering and introduced the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow you to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. Before the continue clause was introduced, route-map configuration was linear and did not allow any control over the flow of a route map.

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map "falls through" and evaluates the next sequence number of the route map until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order starting with the lowest sequence number and ending with the highest sequence number. If the route map contains only set clauses, the set clauses will be executed automatically, and the route map will not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map will continue to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (jump to) a specific route map entry by specifying the sequence number, or if a sequence number is not specified, the continue clause will go to the next sequence number. This behavior is called an "implied continue." If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

Match Operations with Continue Clauses

If a match clause does not exist in the route map entry but a continue clause does, the continue clause will be automatically executed and go to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map entry contains a continue clause, the route map will execute the continue clause if a successful match occurs. If a continue clause does not exist in the next route map entry, the route map will be evaluated normally. If a continue clause exists in the next route map entry but a match does not occur, the route map will not continue and will "fall through" to the next sequence number if one exists.

Note If the number of community lists in a match community clause within a route map exceed 256 characters in a line, you must nvgen multiple match community statements in a new line.

Set Operations with Continue Clauses

Set clauses are saved during the match clause evaluation process and are executed after the route-map evaluation is completed. The set clauses are evaluated and executed in the order in which they were configured. Set clauses are executed only after a successful match occurs, unless the route map does not contain a match clause. The continue statement proceeds to the specified route map entry only after configured set actions are performed. If a set action occurs in the first route map and then the same set action occurs again, with a different value, in a subsequent route map entry, the last set action may override any previous set actions that were configured with the same **set** command unless the **set** command permits more than one value. For example, the **set as-path prepend** command permits more than one autonomous system number to be configured.

Note A continue clause can be executed, without a successful match, if a route map entry does not contain a match clause.

Note Route maps have a linear behavior, not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. For an example, see the "Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map" section.

How to Filter Traffic Using Continue Clauses in a BGP Route Map

Filtering Traffic Using Continue Clauses in a BGP Route Map

Perform this task to filter traffic using continue clauses in a BGP route map.

Note Continue clauses can go only to a higher route map entry (a route map entry with a higher sequence number) and cannot go to a lower route map entry.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** {*ip-address*| *peer-group-name*} **route-map** *map-name* {**in** | **out**}
6. **exit**
7. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
8. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
9. **set community** *community-number* [**additive**] [*well-known-community*] | **none**}
10. **continue** [*sequence-number*]
11. **end**
12. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**neighbor** {*ip-address*| *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in`	Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME permit 10`	Enters route-map configuration mode to create or configure a route map.
Step 8	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Device(config-route-map)# match ip address 1`	Configures a **match** command that specifies the conditions under which policy routing and route filtering occur. • Multiple **match** commands can be configured. If a **match** command is configured, a match must occur in order for the continue statement to be executed. If a **match** command is not configured, set and continue clauses will be executed. **Note** The **match** and **set** commands used in this task are examples that are used to help describe the operation of the **continue** command. For a list of specific **match** and **set** commands, see the **continue** command in the *Cisco IOS IP Routing: BGP Command Reference*.

	Command or Action	Purpose
Step 9	**set community** *community-number* [**additive**] [*well-known-community*] | **none**} **Example:** `Device(config-route-map)# set community 10:1`	Configures a **set** command that specifies the routing action to perform if the criteria enforced by the **match** commands are met. • Multiple **set** commands can be configured. • In this example, a clause is created to set the specified community.
Step 10	**continue** [*sequence-number*] **Example:** `Device(config-route-map)# continue`	Configures a route map to continue to evaluate and execute match statements after a successful match occurs. • If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. • If no sequence number is specified, the continue clause will go to the route map with the next sequence number. This behavior is called an "implied continue." **Note** Continue clauses in outbound route maps are supported in Cisco IOS XE Release 2.1 and later releases.
Step 11	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 12	**show route-map** [*map-name*] **Example:** `Device# show route-map`	(Optional) Displays locally configured route maps. The name of the route map can be specified in the syntax of this command to filter the output.

Examples

The following sample output shows how to verify the configuration of continue clauses using the **show route-map** command. The output displays configured route maps including the match, set, and continue clauses.

```
Device# show route-map

route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
```

```
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  Continue: to next entry 40
  Set clauses:
    as-path prepend 10 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 40
  Match clauses:
    community (community-list filter): 10:1
  Set clauses:
    local-preference 104
  Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
  Match clauses:
  Set clauses:
    community 655370
  Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for BGP Route Map Continue

Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map

The following example shows continue clause configuration in a route map sequence.

Note Continue clauses in outbound route maps are supported only in Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful matches occurs. If a match does not occur, the route map will "fall through" to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries. Only the first successful **match ip address** clause is supported.

If a successful match does not occur in route map entry 20, the route map will fall through to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed and the **continue** clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will fall through to route map entry 30 and execute the set clause. A sequence number is not specified for the **continue** clause, so route map entry 40 will be evaluated.

There are two behaviors that can occur when the same **set** command is repeated in subsequent **continue** clause entries. For **set** commands that configure an additive or accumulative value (for example, **set community additive**, **set extended community additive**, and **set as-path prepend**), subsequent values are added by subsequent entries. The following example illustrates this behavior. After each set of match clauses, a **set as-path prepend** command is configured to add an autonomous system number to the as-path. After a match occurs, the route map stops evaluating match clauses and starts executing the set clauses, in the order in which

they were configured. Depending on how many successful match clauses occur, the as-path is prepended by one, two, or three autonomous system numbers.

```
route-map ROUTE-MAP-NAME permit 10
 match ip address 1
 match metric 10
 set as-path prepend 10
 continue 30
 !
route-map ROUTE-MAP-NAME permit 20
 match ip address 2
 match metric 20
 set as-path prepend 10 10
 !
route-map ROUTE-MAP-NAME permit 30
 set as-path prepend 10 10 10
 continue
 !
route-map ROUTE-MAP-NAME permit 40
 match community 10:1
 set local-preference 104
```

In this example, the same **set** command is repeated in subsequent **continue** clause entries, but the behavior is different from the first example. For **set** commands that configure an absolute value, the value from the last instance will overwrite the previous value(s). The following example illustrates this behavior. The set clause value in sequence 20 overwrites the set clause value from sequence 10. The next hop for prefixes from the 172.16/16 network is set to 10.2.2.2, not 10.1.1.1.

```
ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
 match ip address prefix-list 1
 set ip next hop 10.1.1.1
 continue 20
 exit
route-map RED permit 20
 match ip address prefix-list 2
 set ip next hop 10.2.2.2
 end
```

Note Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. The following example illustrates this case.

In the following example, when routes match an as-path of 10, 20, or 30, the routes are permitted and the continue clause jumps over the explicit deny clause to process the match ip address prefix list. If a match occurs here, the route metric is set to 100. Only routes that do not match an as-path of 10, 20, or 30 and do match a community number of 30 are denied. To deny other routes, you must configure an explicit deny statement.

```
route-map test permit 10
 match as-path 10 20 30
 continue 30
 exit
route-map test deny 20
 match community 30
```

```
 exit
route-map test permit 30
 match ip address prefix-list 1
 set metric 100
 exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

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 Map Continue

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 http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for BGP Route Map Continue

Feature Name	Releases	Feature Information
BGP Route Map Continue	12.3(2)T	The BGP Route Map Continue feature introduces the continue clause to BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configuration need not be repeated within the same route map.