

Configuring Route Policy Manager

This chapter describes how to configure the Route Policy Manager on Cisco NX-OS switches.

This chapter includes the following sections:

- Information About Route Policy Manager, on page 1
- Guidelines and Limitations for Route Policy Manager, on page 5
- Default Settings for Route Policy Manager, on page 6
- Configuring Route Policy Manager, on page 6
- Verifying the Route Policy Manager Configuration, on page 22
- Configuration Examples for Route Policy Manager, on page 22
- Related Topics, on page 22
- Additional References, on page 22

Information About Route Policy Manager

Route Policy Manager supports route maps and IP prefix lists. These features are used for route redistribution. A prefix list contains one or more IPv4 network prefixes and the associated prefix length values. You can use a prefix list by itself in features such as Border Gateway Protocol (BGP) templates, route filtering, or redistribution of routes that are exchanged between routing domains.

Route maps can apply to both routes and IP packets. Route filtering and redistribution pass a route through a route map.

Prefix Lists

You can use prefix lists to permit or deny an address or range of addresses. Filtering by a prefix list involves matching the prefixes of routes or packets with the prefixes listed in the prefix list. An implicit deny is assumed if a given prefix does not match any entries in a prefix list.

You can configure multiple entries in a prefix list and permit or deny the prefixes that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Cisco NX-OS assigns a sequence number automatically. Cisco NX-OS evaluates prefix lists starting with the lowest sequence number. Cisco NX-OS processes the first successful match for a given prefix. Once a match occurs, Cisco NX-OS processes the permit or deny statement and does not evaluate the rest of the prefix list.

Note An empty prefix list permits all routes.

MAC Lists

You can use MAC lists to permit or deny MAC address or range of addresses. A MAC list consists of a list of MAC addresses and optional MAC masks. A MAC mask is a wild-card mask that is logically AND-ed with the MAC address when the route map matches on the MAC list entry. Filtering by a MAC list involves matching the MAC address of packets with the MAC addresses listed in the MAC list. An implicit deny is assumed if a given MAC address does not match any entries in a MAC list.

You can configure multiple entries in a MAC list and permit or deny the MAC addresses that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Cisco NX-OS assigns a sequence number automatically. Cisco NX-OS evaluates MAC lists starting with the lowest sequence number. Cisco NX-OS processes the first successful match for a given MAC address. Once a match occurs, Cisco NX-OS processes the permit or deny statement and does not evaluate the rest of the MAC list.

Route Maps

You can use route maps for route redistribution. Route map entries consist of a list of match and set criteria. The match criteria specify match conditions for incoming routes or packets, and the set criteria specify the action taken if the match criteria are met.

You can configure multiple entries in the same route map. These entries contain the same route map name and are differentiated by a sequence number.

You create a route map with one or more route map entries arranged by the sequence number under a unique route map name. The route map entry has the following parameters:

- Sequence number
- Permission—permit or deny
- Match criteria
- Set changes

By default, a route map processes routes or IP packets in a linear fashion, that is, starting from the lowest sequence number. You can configure the route map to process in a different order using the **continue** statement, which allows you to determine which route map entry to process next.

Match Criteria

You can use a variety of criteria to match a route or IP packet in a route map. Some criteria, such as BGP community lists, are applicable only to a specific routing protocol, while other criteria, such as the IP source or the destination address, can be used for any route or IP packet.

When Cisco NX-OS processes a route or packet through a route map, it compares the route or packet to each of the match statements configured. If the route or packet matches the configured criteria, Cisco NX-OS processes it based on the permit or deny configuration for that match entry in the route map and any set criteria configured.

The match categories and parameters are as follows:

- BGP parameters—Match based on AS numbers, AS-path, community attributes, or extended community attributes.
- Prefix lists—Match based on an address or range of addresses.
- Multicast parameters—Match based on rendezvous point, groups, or sources.
- Other parameters—Match based on IP next-hop address or packet length.

Set Changes

Once a route or packet matches an entry in a route map, the route or packet can be changed based on one or more configured set statements.

The set changes are as follows:

- BGP parameters—Change the AS-path, tag, community, extended community, dampening, local preference, origin, or weight attributes.
- Metrics—Change the route-metric, the route-tag, or the route-type.
- Other parameters—Change the forwarding address or the IP next-hop address.

Access Lists

IP access lists can match the packet to a number of IP packet fields such as the following:

- Source or destination IPv4 address
- Protocol
- Precedence
- ToS

See the Cisco Nexus 3548 Switch NX-OS Security Configuration Guide for more information on ACLs.

AS Numbers for BGP

You can configure a list of AS numbers to match against BGP peers. If a BGP peer matches an AS number in the list and matches the other BGP peer configuration, BGP creates a session. If the BGP peer does not match an AS number in the list, BGP ignores the peer. You can configure the AS numbers as a list, a range of AS numbers, or you can use an AS-path list to compare the AS numbers against a regular expression.

AS-path Lists for BGP

You can configure an AS-path list to filter inbound or outbound BGP route updates. If the route update contains an AS-path attribute that matches an entry in the AS-path list, the router processes the route based on the permit or deny condition configured. You can configure AS-path lists within a route map.

You can configure multiple AS-path entries in an AS-path list by using the same AS-path list name. The router processes the first entry that matches.

Community Lists for BGP

You can filter BGP route updates based on the BGP community attribute by using community lists in a route map. You can match the community attribute based on a community list, and you can set the community attribute using a route map.

A community list contains one or more community attributes. If you configure more than one community attribute in the same community list entry, then the BGP route must match all community attributes listed to be considered a match.

You can also configure multiple community attributes as individual entries in the community list by using the same community list name. In this case, the router processes the first community attribute that matches the BGP route, using the permit or deny configuration for that entry.

You can configure community attributes in the community list in one of the following formats:

- A named community attribute, such as **internet** or **no-export**.
- In **aa:nn** format, where the first two bytes represent the two-byte AS number and the last two bytes represent a user-defined network number.
- A regular expression.

Extended Community Lists for BGP

Extended community lists support 4-byte AS numbers. You can configure community attributes in the extended community list in one of the following formats:

- In *aa4:nn* format, where the first four bytes represent the four-byte AS number and the last two bytes represent a a user-defined network number.
- A regular expression.

Cisco NX-OS supports generic-specific extended community lists, which provide similar functionality to regular community lists for four-byte AS numbers. You can configure generic-specific extended community lists with the following properties:

- Transitive—BGP propagates the community attributes across autonomous systems.
- Nontransitive—BGP removes community attributes before propagating the route to another autonomous system.

Route Redistribution and Route Maps

You can use route maps to control the redistribution of routes between routing domains. Route maps match on the attributes of the routes to redistribute only those routes that pass the match criteria. The route map can also modify the route attributes during this redistribution using the set changes.

The router matches redistributed routes against each route map entry. If there are multiple match statements, the route must pass all of the match criteria. If a route passes the match criteria defined in a route map entry, the actions defined in the entry are executed. If the route does not match the criteria, the router compares the route against subsequent route map entries. Route processing continues until a match is made or the route is processed by all entries in the route map with no match. If the router processes the route against all entries in a route map with no match, the router accepts the route (inbound route maps) or forwards the route (outbound route maps).

Guidelines and Limitations for Route Policy Manager

Route Policy Manager has the following configuration guidelines and limitations:

- Although CLI allows **set** or **match** on **route-tag**, it is not supported and will cause unintended behavior for that particular route-map sequence.
- Names in the prefix-list are case-insensitive. We recommend using unique names. Do not use the same name by modifying upper-case and lower-case characters. For example, CTCPrimaryNetworks and CtcPrimaryNetworks are two different entries.
- If no route map exists, all routes are denied.
- If no prefix list exists, all routes are permitted.
- Without any match statement in a route-map entry, the permission (permit or deny) of the route-map entry decides the result for all the routes or packets.
- If referred policies (for example, prefix lists) within a match statement of a route-map entry return either a no-match or a deny-match, Cisco NX-OS fails the match statement and processes the next route-map entry.
- When you change a route map, Cisco NX-OS holds all the changes until you exit from the route-map configuration submode. Cisco NX-OS then sends all the changes to the protocol clients to take effect.
- Cisco recommends that you do not have both IPv4 and IPv6 match statements in the same route-map sequence. If both are required, they should be specified in different sequences in the same route-map.
- Because you can use a route map before you define it, verify that all your route maps exist when you finish a configuration change.
- You can view the route-map usage for redistribution and filtering. Each individual routing protocol provides a way to display these statistics.
- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map.
- Route Policy Manager does not support MAC lists.
- The maximum number of characters for ACL names in the ip access-list name command is 64. However, ACL names that are associated with RPM commands (such as ip prefix-list and match ip address) accept a maximum of only 63 characters.
- BGP supports only specific **match** commands. For details, see the **match** commands table in the Configuring Route Maps, on page 16 section.
- If you create an ACL named "prefix-list," it cannot be associated with a route map that is created using the match ip address command. The RPM command match ip address prefix-list makes the previous command (with the "prefix-list" ACL name) ambiguous.
- You can configure only one ACL when using the match ip address command.

Default Settings for Route Policy Manager

Table below lists the default settings for Route Policy Manager.

Table 1: Default Route Policy Manager Parameters

Parameters	Default
Route Policy Manager	Enabled

Configuring Route Policy Manager



Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring IP Prefix Lists

IP prefix lists match the IP packet or route against a list of prefixes and prefix lengths. You can create an IP prefix list for IPv4.

You can configure the prefix list entry to match the prefix length exactly, or to match any prefix with a length that matches the configured range of prefix lengths.

Use the **ge** and **lt** keywords to create a range of possible prefix lengths. The incoming packet or route matches the prefix list if the prefix matches and if the prefix length is greater than or equal to the **ge** keyword value (if configured) and less than or equal to the **lt** keyword value (if configured). When using the **eq** keyword, the value you set must be greater than the mask length for the prefix.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **ip prefix-list** *name* **description** *string*
3. **ip prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}]
4. (Optional) **show ip prefix-list** *name*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	(Optional) **ip prefix-list** *name* **description** *string* **Example:** `switch(config)# ip prefix-list AllowPrefix description allows engineering server`	Adds an information string about the prefix list.
Step 3	**ip prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] **Example:** `switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0 eq 24`	Creates an IPv4 prefix list or adds a prefix to an existing prefix list. The prefix length is matched as follows: • eq—Matches the exact *prefix length*. This value must be greater than the mask length. • ge—Matches a prefix length that is equal to or greater than the configured *prefix length*. • le—Matches a prefix length that is equal to or less than the configured *prefix length*.
Step 4	(Optional) **show ip prefix-list** *name* **Example:** `switch(config)# show ip prefix-list AllowPrefix`	Displays information about prefix lists.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to create an IPv4 prefix list with two entries and apply the prefix list to a BGP neighbor:

```
switch# configure terminal
switch(config)# ip prefix-list allowprefix seq 10 permit 192.0.2.0/23 eq 24
switch(config)# ip prefix-list allowprefix seq 20 permit 209.165.201.0/27 eq 28
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

Configuring MAC Lists

You can configure a MAC list to permit or deny a range of MAC addresses.

SUMMARY STEPS

1. **configure terminal**
2. **mac-list** *name* [**seq** *number*] { **permit** | **deny** } *mac-address* {*mac-mask*]}
3. (Optional) **show mac-list** *name*
4. (Optional) **show ip prefix-list** *name*

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**mac-list** *name* [**seq** *number*] { **permit** | **deny** } *mac-address* {*mac-mask*]} **Example:** `switch(config)# mac-list AllowMac seq 1 permit 0022.5579.a4c1 ffff.ffff.0000`	Creates a MAC list or adds a MAC address to an existing MAC list. The *seq* range is from 1 to 4294967294. The *mac-mask* specifies the portion of the MAC address to match against and is in MAC address format.
Step 3	(Optional) **show mac-list** *name* **Example:** `switch(config)# show mac-list AllowMac`	Displays information about MAC lists.
Step 4	(Optional) **show ip prefix-list** *name* **Example:** `switch(config)# show ip prefix-list AllowPrefix`	Displays information about prefix lists.

Configuring AS-path Lists

You can specify an AS-path list filter on both inbound and outbound BGP routes. Each filter is an access list based on regular expressions. If the regular expression matches the representation of the AS-path attribute of the route as an ASCII string, then the permit or deny condition applies.

SUMMARY STEPS

1. **configure terminal**
2. **ip as-path access-list** *name* { **deny** | **permit** } *expression*
3. (Optional) **show ip as-path-access-list** *name*
4. (Optional) **show ip prefix-list** *name*

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**ip as-path access-list** *name* { **deny** | **permit** } *expression* **Example:** `switch(config)# ip as-path access-list Allow40 permit 40`	
Step 3	(Optional) **show ip as-path-access-list** *name* **Example:** `switch(config)# show ip as-path-access-list Allow40`	Displays information about as-path access lists.
Step 4	(Optional) **show ip prefix-list** *name* **Example:** `switch(config)# show ip prefix-list AllowPrefix`	Displays information about prefix lists.

Example

This example shows how to create an AS-path list with two entries and apply the AS path list to a BGP neighbor:

```
switch# configure terminal
switch(config)# ip as-path access-list AllowAS permit 64510
switch(config)# ip as-path access-list AllowAS permit 64496
switch(config)# copy running-config startup-config
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# filter-list AllowAS in
```

Replacing BGP AS-path Attribute

The following procedures allow you to manipulate the BGP routing policy by modifying the BGP as-path attribute in inbound and outbound route maps.

Consider the following guidelines when replacing the BGP as-path attribute:

- This feature is applicable to only eBGP neighbors on a per address family identifier (AFI) basis. If you attempt to configure the feature on iBGP neighbors, the configuration is ignored.
- A route map with this feature can be applied to both the inbound and outbound sides of a BGP neighbor.

- This feature supports any combination of AS_SET, AS_SEQUENCE, CONFED_SET, and CONFED_SEQUENCE.
- When interacting with a BGP speaker that supports only a 2-byte AS, the 4-byte AS number is replaced by the reserved 2-byte AS number 23456.
- If a confederation indentifier is configured, consider using the confederation indentifier as the local ASN in the CLI when interacting with a peer that is outside the confederation. When interacting with a peer belonging to the same confederation, consider using the process ASN in the **router bgp** *asn* command.
- When the BGP **local-as** feature is configured, the configured local-as will be considered as local ASN in the CLI.
- For outbound route-maps, the local ASN will always be prepended to the resulting as_path from the CLI.
- A maximum of 32 AS numbers can be configured in a **set as-path** or **set as-path replace** command.
- Only one of these options can be configured under one route-map sequence: **set as-path**, **set as-path prepend**, and **set as-path replace**.
- If **remove-private-as** is configured, it will be applied before applying the new route-map commands on the outbound side.
- If **as-override** is configured, it will be applied after applying the new route-map commands on the outbound side.
- AS_PATH loop checks will execute on the original AS_PATH before the new route-map commands are applied on both inbound and outbound sides. These checks can be relaxed by using **allow-as in** on the inbound side and **disable-peer-as-check** on the outbound side.

Replacing the Complete AS-path

Use this procedure to modify the AS-path in an incoming or outgoing BGP update to a custom AS-path. You can also remove the AS-path completely.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	Example:
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map.

Step 3	[**no**] **set as-path** { **none** | {*as-number* | **remote-as** | **local-as**}+] } **Example:**	Replaces AS_PATH with a list of custom ASNs or clears the AS_PATH. The command options are: • *as-number*: The specified AS number.

	Command or Action	Purpose
	`switch(config-route-map)# set as-path 11 local-as remote-as 13`	• **remote-as**: The AS number of the BGP peer. • **local-as**: The local AS number. The **none** keyword removes the AS-path completely.

Example

In the following examples, these values are assumed:

- The original AS_PATH is **10 20 30 40 50 60**.
- The local-as is **100**.
- The remote-as is **200**.

This example shows how to specify a custom AS-path. This command will change the AS-path to **11 100 200 13 200 10.10 65535**.

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path 11 local-as remote-as 13 remote-as 10.10 65535
```

This example shows how to clear the AS-path. This command will cause the AS-path to be empty.

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path none
```

Replacing Selected AS Numbers in the AS-path

Use this procedure to replace specific AS numbers in the AS-path and replace them with custom AS numbers in an incoming or outgoing BGP update. You can also specify **private-as** as a match keyword. In this case, any instance of a private-as is matched and can be replaced or removed.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	Example:
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map.

	Command or Action	Purpose
Step 3	[**no**] **set as-path replace** {*asn_list* | **private-as**} [**with** {*as-number* | **remote-as** | **none**}] **Example:** `switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as`	If the **with** keyword is not specified, substitute the local-as for any instance of an ASN mentioned in the comma separated *asn_list*, or for any private-as if the **private-as** keyword is specified. If the **with** keyword is specified, substitute the value after the **with** keyword for any matched ASN, or any private-as if the **private-as** keyword is specified. The command options following the **with** keyword are: • *as-number*: The matched values are replaced by the specified AS number. • **remote-as**: The matched values are replaced by the AS number of the BGP peer. • **none**: The matched values are removed from the AS-path.

Example

In the following examples, these values are assumed:

- The original AS_PATH is **1 5 2 10.10 65534 20**.
- The local-as is **100**.
- The remote-as is **200**.

This example shows how to replace two specific ASNs and a private-as with the local-as. This command will change the AS-path to **100 5 100 10.10 100 20**.

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as
```

This example shows how to replace two specific ASNs and a private-as with the neighbor's ASN (remote-as). This command will change the AS-path to **200 5 200 10.10 200 20**.

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as
```

This example shows how to remove two specific ASNs and a private-as. This command will change the AS-path to **5 10.10 20**.

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with none
```

Configuring Community Lists

You can use community lists to filter BGP routes based on the community attribute. The community number consists of a 4-byte value in the *aa:nn* format. The first two bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same community list statement, all community values must match to satisfy the community list filter. When you configure multiple values in separate community list statements, the first list that matches a condition is processed.

Use community lists in a match statement to filter BGP routes based on the community attribute.

SUMMARY STEPS

1. **configure terminal**
2.
3. (Optional) show ip community-list *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2		

Option	Description
Command	Purpose
ip community-list standard *list-name* { **deny** | **permit** } [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**] Example: `switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20`	Creates a standard BGP community list. The *list-name* can be any case-sensitive, alphanumeric string up to 63 characters. The *community-list* can be one or more communities in the *aa:nn* format.
ip community-list expanded *list-name* { **deny** | **permit** } *expression* Example: `switch(config)# ip community-list expanded`	Creates an expanded BGP community list using a regular expression.

	Command or Action		Purpose
	Option	**Description**	
	`BGPComplex deny 50000:[0-9][0-9]_`		
Step 3	(Optional) show ip community-list *name* **Example:** `switch(config)# show ip community-list BGPCommunity`		Displays information about community lists.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`		Saves this configuration change.

Example

This example shows how to create a community list with two entries:

```
switch# configure terminal
switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20
switch(config)# ip community-list standard BGPCommunity permit local-AS no-export
switch(config)# copy running-config startup-confi
```

Configuring Extended Community Lists

You can use extended community lists to filter BGP routes based on the community attribute. The community number consists of a 6-byte value in the *aa4:nn* format. The first four bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same extended community list statement, all extended community values must match to satisfy the extended community list filter. When you configure multiple values in separate extended community list statements, the first list that matches a condition is processed.

Use extended community lists in a match statement to filter BGP routes based on the extended community attribute.

SUMMARY STEPS

1. **configure terminal**
2.
3. (Optional) show ip community-list *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td>configure terminal
Example:
<pre>switch# configure terminal
switch(config)#</pre></td><td>Enters configuration mode.</td></tr>
<tr><td>Step 2</td><td><table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Command</td><td>Purpose</td></tr>
<tr><td>ip community-list standard list-name { deny | permit } 4bytegeneric { transitive | non-transitive } community1 [community2]
Example:
<pre>switch(config)# ip extcommunity-list standard BGPExtCommunity permit 4bytegeneric transitive 65536:20</pre></td><td>Creates a standard BGP community list. The community-list can be one or more extended communities in the aa:nn format.</td></tr>
<tr><td>ip extcommunity-list expanded list-name { deny | permit } expression
Example:
<pre>switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]_</pre></td><td>Creates an expanded BGP community list using a regular expression.</td></tr>
</table></td><td></td></tr>
<tr><td>Step 3</td><td>(Optional) show ip community-list name
Example:
<pre>switch(config)# show ip community-list BGPCommunity</pre></td><td>Displays information about community lists.</td></tr>
<tr><td>Step 4</td><td>(Optional) copy running-config startup-config
Example:
<pre>switch# copy running-config startup-config</pre></td><td>Saves this configuration change.</td></tr>
</table>

Example

This example shows how to create a generic-specific extended community list:

```
switch# configure terminal
switch(config)# ip extcommunity-list standard test1 permit 4bytegeneric transitive 65536:40
 65536:60
switch(config)# copy running-config startup-config
```

Configuring Route Maps

Before you begin

You can use route maps for route redistribution or route filtering. Route maps can contain multiple match criteria and multiple set criteria.

Configuring a route map for BGP triggers an automatic soft clear or refresh of BGP neighbor sessions.

SUMMARY STEPS

1. **configure terminal**
2. configure terminal
3. (Optional) **continue** *seq*
4. (Optional) **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	configure terminal
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 3	(Optional) **continue** *seq* **Example:** `switch(config-route-map)# continue 10`	Determines what sequence statement to process next in the route map. Used only for filtering and redistribution.
Step 4	(Optional) **exit** **Example:** `switch(config-route-map)# continue 10`	Determines what sequence statement to process next in the route map. Used only for filtering and redistribution.
Step 5	(Optional) **copy running-config startup-config** **Example:**	Saves this configuration change.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	

Example

You can configure the following optional match parameters for route maps in route-map configuration mode:



Note The **default-information originate** command ignores **match** statements in the optional route map.

Command	Purpose
match as-path *name* [*name...*] **Example:** `switch(config-route-map)# match as-path Allow40`	Matches against one or more AS-path lists. Create the AS-path list with the **ip as-path access-list** command.
match as-number { *number* [,*number...*] | **as-path-list** *name* [*name...*]} **Example:** `switch(config-route-map)# match as-number 33,50-60`	Matches against one or more AS numbers or AS-path lists. Create the AS-path list with the **ip as-path access-list** command. The number range is from 1 to 65535. The AS-path list name can be any case-sensitive, alphanumeric string up to 63 characters.
match community name [*name...*] [**exact-match**] **Example:** `switch(config-route-map)# match community BGPCommunity`	Matches against one or more community lists. Create the community list with the **ip community-list** command.
match extcommunity *name* [*name...*] [**exact-match**] **Example:** `switch(config-route-map)# match extcommunity BGPextCommunity`	Matches against one or more extended community lists. Create the community list with the **ip extcommunity-list** command.
match interface *interface-type number* [*interface-type number...*] **Example:** `switch(config-route-map)# match interface e 1/2`	Matches any routes that have their next hop out one of the configured interfaces. Use **?** to find a list of supported interface types.
match ip address prefix-list *name* [*name...*] **Example:** `switch(config-route-map)# match ip address prefix-list AllowPrefix`	Matches against one or more IPv4 prefix lists. Use the **ip prefix-list** command to create the prefix list.

Command	Purpose
match ip next-hop prefix-list *name* [*name* ...] **Example:** `switch(config-route-map)# match ip next-hop prefix-list AllowPrefix`	Matches the IPv4 next-hop address of a route to one or more IP prefix lists. Use the *ip prefix-list* command to create the prefix list.
match ip route-source prefix-list *name* [*name* ...] **Example:** `switch(config-route-map)# match ip route-source prefix-list AllowPrefix`	Matches the IPv4 route source address of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list.
match mac-list *name* [*name*...] **Example:** `switch(config-route-map)# match mac-list AllowMAC`	Matches against one or more MAC lists. Use the **mac-list** command to create the MAC list.
match metric *value* [*+-deviation*] [*value..*] **Example:** `switch(config-route-map)# match mac-list AllowMAC`	Matches the route metric against one or more metric values or value ranges. Use +- **deviation** argument to set a metric range. The route map matches any route metric that falls the range: *value* - *deviation* to *value* + *deviation*.
match route-type *route-type* **Example:** `switch(config-route-map)# match route-type level 1 level 2`	Matches against a type of route. The *route-type* can be one or more of the following: • external • internal • level-1 • level-2 • local • nssa-external • type-1 • type-2
match tag *tagid* [*tagid...*] **Example:** `switch(config-route-map)# match tag 2`	Matches a route against one or more tags for filtering or redistribution.
match vlan *vlan-id* [*vlan-range*] **Example:** `switch(config-route-map)# match vlan 3, 5-10`	Matches against a VLAN.

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	Purpose
set as-path { **tag** | **prepend** { **last-as** *number* | *as-1* [*as-2...*]}} **Example:** `switch(config-route-map)# set as-path prepend 10 100 110`	Modifies an AS-path attribute for a BGP route. You can prepend the configured*number* of last AS numbers or a string of particular AS-path values (*as-1 as-2...as-n*).
set comm-list *name* **delete** **Example:** `switch(config-route-map)# set comm-list BGPCommunity delete`	Removes communities from the community attribute of an inbound or outbound BGP route update. Use the **ip community-list** command to create the community list.
set community { **none** | **additive** | **local-AS** | **no-advertise** | **no-export** | *community-1* [*community-2...*]} **Example:** `switch(config-route-map)# set community local-AS`	Sets the community attribute for a BGP route update. **Note** When you use both the **set community** and **set comm-list delete** commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. **Note** Use the **send-community** command in BGP neighbor address family configuration mode to propagate BGP community attributes to BGP peers.
set dampening *halflife reuse suppress duration* **Example:** `switch(config-route-map)# set dampening 30 1500 10000 120`	Sets the following BGP route dampening parameters: • *halflife* —The range is from 1 to 45 minutes. The default is 15. • *reuse* —The range is from is 1 to 20000 seconds. The default is 750. • *suppress* —The range is from is 1 to 20000. The default is 2000. • *duration* —The range is from is 1 to 255 minutes. The default is 60.
set extcomm-list *name* **delete** **Example:** `switch(config-route-map)# set extcomm-list BGPextCommunity delete`	Removes communities from the extended community attribute of an inbound or outbound BGP route update. Use the **ip extcommunity-list** command to create the extended community list.

Command	Purpose
set extcommunity generic { **transitive** | **nontransitive** }{ **none** | **additive**] *community-1* [*community-2...*]} **Example:** `switch(config-route-map)# set extcommunity generic transitive 1.0:30`	Sets the extended community attribute for a BGP route update. **Note** When you use both the **set extcommunity** and **set extcomm-list delete** commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. **Note** Use the **send-community** command in BGP neighbor address family configuration mode to propagate BGP extended community attributes to BGP peers.
set forwarding-address **Example:** `switch(config-route-map)# set forwarding-address`	Sets the forwarding address for OSPF.
set level { **backbone** | **level-1** | **level-1-2** | **level-2** } **Example:** `switch(config-route-map)# set level backbone`	Sets what area to import routes to for IS-IS. The options for IS-IS are level-1, level-1-2, or level-2. The default is level-1.
set local-preference *value* **Example:** `switch(config-route-map)# set local-preference 4000`	Sets the BGP local preference value. The range is from 0 to 4294967295.
set metric [+ | -] *bandwidth-metric* **Example :** `switch(config-route-map)# set metric +100`	Adds or subtracts from the existing metric value. The metric is in Kb/s. The range is from 0 to 4294967295.

Command	Purpose
set metric *bandwidth* [*delay reliability load mtu*] **Example :** `switch(config-route-map)# set metric 33 44 100 200 1500`	Sets the route metric values. Metrics are as follows: • *metric0* —Bandwidth in Kb/s. The range is from 0 to 4294967295. • *metric1* —Delay in 10-microsecond units. • *metric2* —Reliability. The range is from 0 to 255 (100 percent reliable). • *metric3* —Loading. The range is from 1 to 200 (100 percent loaded). • *metric4* —MTU of the path. The range is from 1 to 4294967295.
set metric-type { **external** | **internal** | **type-1** | **type-2** } **Example:** `switch(config-route-map)# set metric-type internal`	Sets the metric type for the destination routing protocol. The options are as follows: external—IS-IS external metric internal— IGP metric as the MED for BGP type-1—OSPF external type 1 metric type-2—OSPF external type 2 metric
set origin { **egp** *as-number* | **igp** | **incomplete**} **Example:** `switch(config-route-map)# set origin incomplete`	Sets the BGP origin attribute. The EGP *as-number* range is from 0 to 65535.
set tag *name* **Example:** `switch(config-route-map)# set tag 33`	Sets the tag value for the destination routing protocol. The *name* parameter is an unsigned integer.
set weight *count* **Example:** `switch(config-route-map)# set weight 33`	Sets the weight for the BGP route. The range is from 0 to 65535.

The **set metric-type internal** command affects an outgoing policy and an eBGP neighbor only. If you configure both the **metric** and **metric-type internal** commands in the same BGP peer outgoing policy, then Cisco NX-OS ignores the **metric-type internal** command.

Verifying the Route Policy Manager Configuration

To display the route policy manager configuration information, perform one of the following tasks:

Command	Purpose
show ip community-list [*name*]	Displays information about a community list.
show ip ext community-list [*name*]	Displays information about an extended community list.
show [ip] prefix-list [*name*]	Displays information about an IPv4 prefix list.
show route-map [*name*]	Displays information about a route map.

Configuration Examples for Route Policy Manager

This example shows how to use an address family to configure Route Policy Manager so that any unicast and multicast routes from neighbor 172.16.0.1 are accepted if they match prefix-list AllowPrefix:

```
router bgp 64496

neighbor 172.16.0.1 remote-as 64497
  address-family ipv4 unicast
    route-map filterBGP in

route-map filterBGP
 match ip address prefix-list AllowPrefix

ip prefix-list AllowPrefix 10 permit 192.0.2.0/24
ip prefix-list AllowPrefix 20 permit 172.16.201.0/27
```

Related Topics

The following topics can give more information on Route Policy Manager:

- Configuring Basic BGP

Additional References

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

- Related Documents
- Standards

Related Documents

Related Topic	Document Title
Route Policy Manager CLI commands	Cisco Nexus 3000 Series Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—