



# Routing Policy Language Commands

---

This module describes the Cisco IOS XR software routing policy language (RPL) commands used to create, modify, monitor, and maintain routing policies.

For detailed information about RPL concepts, configuration tasks, and examples, see the *Implementing Routing Policy on Cisco IOS XR Software* module in the *Cisco IOS XR Routing Configuration Guide for the Cisco XR 12000 Series Router*.

- [abort \(RPL\), page 6](#)
- [add, page 8](#)
- [apply, page 10](#)
- [as-path in, page 12](#)
- [as-path is-local, page 14](#)
- [as-path length, page 16](#)
- [as-path neighbor-is, page 18](#)
- [as-path originates-from, page 21](#)
- [as-path passes-through, page 24](#)
- [as-path-set, page 27](#)
- [as-path unique-length, page 29](#)
- [community is-empty, page 31](#)
- [community matches-any, page 32](#)
- [community matches-every, page 35](#)
- [community-set, page 38](#)
- [delete community, page 41](#)
- [delete extcommunity rt, page 43](#)
- [destination in, page 45](#)
- [done, page 47](#)
- [drop, page 49](#)

- [edit](#), page 51
- [end-global](#), page 54
- [end-policy](#), page 55
- [end-set](#), page 57
- [extcommunity rt is-empty](#), page 59
- [extcommunity rt matches-any](#), page 61
- [extcommunity rt matches-every](#), page 63
- [extcommunity rt matches-within](#), page 65
- [extcommunity-set cost](#), page 67
- [extcommunity-set rt](#), page 69
- [extcommunity-set soo](#), page 71
- [extcommunity soo is-empty](#), page 73
- [extcommunity soo matches-any](#), page 75
- [extcommunity soo matches-every](#), page 77
- [globalVarN is](#), page 79
- [if](#), page 81
- [local-preference](#), page 88
- [med](#), page 90
- [next-hop in](#), page 92
- [orf prefix in](#), page 94
- [origin is](#), page 96
- [pass](#), page 98
- [path-type is](#), page 100
- [policy-global](#), page 102
- [prefix-set](#), page 104
- [prepend as-path](#), page 107
- [protocol](#), page 109
- [rd in](#), page 111
- [rd-set](#), page 113
- [replace as-path](#), page 115
- [rib-has-route](#), page 117
- [route-has-label](#), page 119
- [route-policy \(RPL\)](#), page 120

- [route-type is, page 122](#)
- [rpl editor, page 124](#)
- [rpl maximum, page 126](#)
- [set community, page 128](#)
- [set core-tree, page 130](#)
- [set dampening, page 132](#)
- [set eigrp-metric, page 134](#)
- [set extcommunity cost, page 136](#)
- [set extcommunity rt, page 138](#)
- [set ip-precedence, page 140](#)
- [set isis-metric, page 142](#)
- [set label, page 143](#)
- [set label-mode, page 144](#)
- [set level, page 146](#)
- [set local-preference, page 148](#)
- [set med, page 150](#)
- [set metric-type \(IS-IS\), page 152](#)
- [set metric-type \(OSPF\), page 154](#)
- [set next-hop, page 156](#)
- [set origin, page 158](#)
- [set ospf-metric, page 160](#)
- [set qos-group \(RPL\), page 161](#)
- [set rib-metric, page 163](#)
- [set rip-metric, page 164](#)
- [set rip-tag, page 166](#)
- [set rpf-topology, page 167](#)
- [set rtset route-limit , page 169](#)
- [set spf-priority, page 171](#)
- [set tag, page 173](#)
- [set traffic-index, page 175](#)
- [set vpn-distinguisher, page 177](#)
- [set weight, page 179](#)
- [show rpl, page 181](#)

- [show rpl active as-path-set, page 183](#)
- [show rpl active community-set, page 186](#)
- [show rpl active extcommunity-set, page 189](#)
- [show rpl active prefix-set, page 192](#)
- [show rpl active rd-set, page 195](#)
- [show rpl active route-policy, page 197](#)
- [show rpl as-path-set, page 200](#)
- [show rpl as-path-set attachpoints, page 202](#)
- [show rpl as-path-set references, page 205](#)
- [show rpl community-set, page 208](#)
- [show rpl community-set attachpoints, page 210](#)
- [show rpl community-set references, page 213](#)
- [show rpl extcommunity-set, page 216](#)
- [show rpl inactive as-path-set, page 220](#)
- [show rpl inactive community-set, page 223](#)
- [show rpl inactive extcommunity-set, page 226](#)
- [show rpl inactive prefix-set, page 229](#)
- [show rpl inactive rd-set, page 232](#)
- [show rpl inactive route-policy, page 234](#)
- [show rpl maximum, page 237](#)
- [show rpl policy-global references, page 239](#)
- [show rpl prefix-set, page 241](#)
- [show rpl prefix-set attachpoints, page 243](#)
- [show rpl prefix-set references, page 246](#)
- [show rpl rd-set, page 249](#)
- [show rpl rd-set attachpoints, page 251](#)
- [show rpl rd-set references, page 253](#)
- [show rpl route-policy, page 256](#)
- [show rpl route-policy attachpoints, page 259](#)
- [show rpl route-policy inline, page 262](#)
- [show rpl route-policy references, page 264](#)
- [show rpl route-policy uses, page 267](#)
- [show rpl unused as-path-set, page 270](#)

- [show rpl unused community-set, page 273](#)
- [show rpl unused extcommunity-set, page 276](#)
- [show rpl unused prefix-set, page 278](#)
- [show rpl unused rd-set, page 281](#)
- [show rpl unused route-policy, page 283](#)
- [source in, page 286](#)
- [suppress-route, page 288](#)
- [tag, page 290](#)
- [tag in, page 292](#)
- [tag-set, page 294](#)
- [unsuppress-route, page 296](#)
- [var globalVarN, page 298](#)
- [vpn-distinguisher is, page 300](#)

# abort (RPL)

To discard a route policy or set definition and return to global configuration mode, use the **abort** command in the appropriate configuration mode.

**abort**

## Syntax Description

This command has no keywords or arguments.

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

Prefix set configuration

Route distinguisher set configuration

AS path set configuration

Community set configuration

Extended community set configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows how to discard a route policy definition that was started and return to global configuration mode:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# if as-path is-local then
RP/0/0/CPU0:router(config-rpl-if)# abort
RP/0/0/CPU0:router(config)#
```

The following example shows how to discard a prefix set definition that was started and return to global configuration mode:

```
RP/0/0/CPU0:router(config)# prefix-set legal-ipv4-prefix-examples  
RP/0/0/CPU0:router(config-pfx)# 10.0.1.1,  
RP/0/0/CPU0:router(config-pfx)# 10.0.2.0/24,  
RP/0/0/CPU0:router(config-pfx)# abort  
RP/0/0/CPU0:router(config)#
```

# add

To add a value to an Routing Information Protocol (RIP) or Enhanced Interior Gateway Protocol (EIGRP) existing metric, use the **add** command in route-policy configuration mode.

**add** {**eigrp-metric** *bandwidth delay reliability loading max-transmission*| **rip-metric** {*number*| *parameter*}}

## Syntax Description

<b>eigrp-metric</b>	Specifies an EIGRP metric attribute.
<i>bandwidth</i>	Bandwidth in kilobits per second. Range is from 0 to 4294967295.
<i>delay</i>	Delay in 10-microsecond units. Range is from 0 to 4294967295.
<i>reliability</i>	Reliability metric. 255 is 100-percent reliable. Range is from 0 to 255.
<i>loading</i>	Effective bandwidth (loading). 255 is 100-percent loaded. Range is from 0 to 255.
<i>max-transmission</i>	Maximum transmission of the path. Range is from 0 to 65535.
<b>rip-metric</b>	Specifies an RIP metric attribute.
<i>number</i>	Value assigned to a four-bit unsigned integer. Range is from 0 to 16.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

If the add value is greater than the maximum allowed value, the metric is added. If the resulting metric exceeds the maximum for the routing protocol, then the route is dropped (by the client routing protocol).



**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following example shows how to offset the RIP metric value:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# add rip-metric 4
RP/0/0/CPU0:router(config-rpl)# end-policy
```

The following example shows how to set the EIGRP metric value:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# add eigrp-metric 50000 24000 230 14000
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# apply

To execute a parameterized or unparameterized policy from within another policy, use the **apply** command in route-policy configuration mode.

**apply** *policy\_name* [*argument1*, *argument2*, . . . , *argumentN*]

## Syntax Description

<i>policy_name</i>	Name of a route policy.
<i>argument</i>	(Optional) Parameter name. The <i>argument</i> can be a value (for example, '100' ) or a parameter (for example, '\$parameter')

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 4.2.1	Wildcard support was added for apply policy-names.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **apply** command to execute a policy (either parameterized or unparameterized) from within another policy, which allows for the reuse of common blocks of policy.

Wildcards can be used in apply policy names. This supports the nested wildcard apply scenario. A wildcard is specified by inserting an asterisk (\*) in place of one of the portions of the apply policy name; the wildcard indicates that any value for that portion of the apply policy name matches. The nested wildcard apply policy allows wildcard (\*) based apply nesting. The wildcard operation permits declaration of a generic apply statement that calls all policies that contain a specific defined set of alphanumeric characters, defined on the router.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the policy CustomerIn applies the route-policy SetLocalPref to conditionally set the local preference on a route. The parameters 20, 30, 40, and 50 are passed to the parameterized policy SetLocalPref, where the local preference is set to:

- 20, if the community 217:20 is present in the route
- 30, if the community 217:30 is present in the route
- 40, if the community 217:40 is present in the route
- 50, if the community 217:50 is present in the route

```
RP/0/0/CPU0:router(config)# route-policy SetLocalPref ($lp0, $lp1, $lp2, $lp3, $lp4)
RP/0/0/CPU0:router(config-rpl)# if community matches-any ($lp0:$lp1) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference $lp1
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-any ($lp0:$lp2) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference $lp2
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-any ($lp0:$lp3) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference $lp3
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-any ($lp0:$lp4) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference $lp4
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy

RP/0/0/CPU0:router(config)# route-policy CustomerIn($cust)
RP/0/0/CPU0:router(config-rpl)# apply SetLocalPref ($cust, 20, 30, 40, 50)
RP/0/0/CPU0:router(config-rpl)# end-policy

RP/0/0/CPU0:router(config)# route-policy Cust_217
RP/0/0/CPU0:router(config-rpl)# apply CustomerIn(217)
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# as-path in

To match the AS path of a route to an AS path set, use the **as-path in** command in route-policy configuration mode.

**as-path in** {*as-path-set-name*|*inline-as-path-set*|*parameter*}

## Syntax Description

<i>as-path-set-name</i>	Name of an AS path set.
<i>inline-as-path-set</i>	Inline AS path set. The inline AS path set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path in** command as a conditional expression within an **if** statement to match the AS path of a route to an AS path set. The AS path is a sequence of autonomous system numbers traversed by a route.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **as-path in** command evaluates to true if at least one of the regular expressions defined in the associated AS path set matches the AS path attribute of the route.

In the case where the AS path set is defined but contains no elements in it, the **as-path in** conditional expression command returns false.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

For example, assume we have an AS path set named my-as-set defined as follows:

```
RP/0/0/CPU0:router(config)# as-path-set my-as-set
RP/0/0/CPU0:router(config-as)# ios-regex '_12$',
RP/0/0/CPU0:router(config-as)# ios-regex '_13$'
RP/0/0/CPU0:router(config-as)# end-set
```

and the following policy excerpt using an *as-path-set-name* argument:

```
RP/0/0/CPU0:router(config-rpl)# if as-path in my-as-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

The AS path in condition evaluates to true if one or more of the regular expression matches in the set my-as-set match the AS path associated with the route. In the case of a defined but empty AS path set, this operator returns false.

The preceding policy excerpt is equivalent to the following version, which uses an *inline-as-path* set variable:

```
RP/0/0/CPU0:router(config-rpl)# if as-path in (ios-regex '_12$',ios-regex '_13$') then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

# as-path is-local

To determine if this router or another router within this autonomous system or confederation originated a Border Gateway Protocol (BGP) route, use the **as-path is-local** command in route-policy configuration mode.

**as-path is-local**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was supported.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path is-local** command as a conditional expression within an **if** statement to determine if this router (or another router within this autonomous system or confederation) originated the route.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

Routes that are locally originated within the autonomous system or confederation carry an empty AS path. For the Border Gateway Protocol (BGP) specification, when a route is advertised across the autonomous system boundary or a confederation boundary, the local autonomous system number or confederation ID is appended to the autonomous system path. The AS path of a locally originated aggregate is also empty unless it has been modified by policy.

The **is-local** operator evaluates to true for autonomous system paths that are empty. An empty AS path is how an AS path that is local to our autonomous system is represented in BGP.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, if the AS path is local, then the local preference is set to 100:

```
RP/0/0/CPU0:router(config-rpl)# if as-path is-local then  
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

## as-path length

To compare the number of ASN in the AS path of a Border Gateway Protocol (BGP) route, use the **as-path length** command in route-policy configuration mode.

**as-path length** {eq| is| ge| le} {*number*| *parameter*}

### Syntax Description

<b>eq   is   ge   le</b>	Equal to; greater than or equal to; less than or equal to.
<i>number</i>	Value assigned to an 11-bit unsigned integer. Range is from 0 to 2047.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path length** command as a conditional expression within an **if** statement to perform a conditional check based on the length of the AS path.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command takes either a specific integer value or a range of integer values specified with the **ge** and **le** operators. Any or all these integers can be parameterized. The operator counts one for each autonomous system in the path. In the case where the route may be aggregated and contain one or more AS sets, the length operator adds one for each set present, the occurrence of an AS set typically indicates that this route is an aggregated route, and the aggregated route had a component route that contained one of the autonomous systems in the set. Likewise, in the case of confederations, a count of one is added for each confederation in the path or each confederation set in the path. A null AS path has a length of zero.



**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, if the AS path length equals 10, then the local preference is set to 100:

```
RP/0/0/CPU0:router(config-rpl)# if as-path length eq 10 then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

**Related Commands**

Command	Description
<a href="#">as-path in, on page 12</a>	Matches the AS path of a route to an AS path set.
<a href="#">as-path originates-from, on page 21</a>	Compares an AS path against the AS sequence beginning with the AS number that originated a route
<a href="#">as-path passes-through, on page 24</a>	Verifies if the supplied integer or parameter appears anywhere in the AS path or if the supplied sequence of integers and parameters appears, in the same order, anywhere in the AS path.
<a href="#">as-path unique-length, on page 29</a>	Performs specific checks based on the length of the AS path.

## as-path neighbor-is

To test autonomous system numbers at the head of the AS path against a sequence of one or more values or parameters, use the **as-path neighbor-is** command in route-policy configuration mode.

**as-path neighbor-is** *as-number-list* [**exact**]

### Syntax Description

<i>as-number-list</i>	Numbers or parameters, enclosed in single quotation marks, that represent a sequence of autonomous system numbers. <ul style="list-style-type: none"> <li>• Range for 2-byte Autonomous system numbers (ASNs) is 1 to 65535.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asplain format is 1 to 4294967295.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asdot format is 1.0 to 65535.65535.</li> </ul>
<b>exact</b>	(Optional) Specifies that with the <b>exact</b> keyword, the <i>as-number-list</i> value must identically match the AS path for the route; without the <b>exact</b> keyword, any element in the <i>as-number-list</i> argument matches one or more occurrences of that element in the AS path for the route.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>as-number-list</i> 4-byte number range 1.0 to 65535.65535 was supported.
Release 3.9.0	Asplain format for 4-byte Autonomous system numbers notation was supported.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path neighbor-is** command as a conditional expression within an **if** statement to test the autonomous system number or numbers at the head of the AS path against a sequence of one or more integral

values or parameters. In other words, to test to learn if the sequence of autonomous system numbers matches the path beginning with the neighboring autonomous system from which this route was heard.

**Note**

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command has an equivalent regular expression (ios-regex). For example, AS path neighbor-is '1' would be '^1\_'.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following are incomplete configuration examples:

```
RP/0/0/CPU0:router(config-rpl)# if as-path neighbor-is '10' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path neighbor-is '$asnum' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path neighbor-is '10 20' then
```

These statements evaluate to true when the first autonomous system numbers on the AS path match, in the same order, the supplied parameters or integer values in the **neighbor-is** statement. If the neighboring autonomous system location happens to be an AS-set, the operator evaluates to true if the corresponding argument to the **neighbor-is** operator is an element of the AS-set.

Without the **exact** keyword, repeated autonomous system numbers in the AS path are ignored. For example,

```
RP/0/0/CPU0:router(config-rpl)# if as-path neighbor-is '10 20' then
```

matches an AS path beginning

```
10 10 10 20 ...
```

and an AS path beginning:

```
10 20 ....
```

With the **exact** keyword, repetitions are not ignored, therefore

```
RP/0/0/CPU0:router(config-rpl)# if as-path neighbor-is '10 20' exact then
```

matches the second of these AS paths but not the first.

**Related Commands**

Command	Description
<a href="#">as-path in, on page 12</a>	Matches the AS path of a route to an AS path set.
<a href="#">as-path length, on page 16</a>	Compares the number of ASN in the AS path of a route.

Command	Description
<a href="#">as-path originates-from, on page 21</a>	Compares an AS path to the AS sequence beginning with the AS number that originated a route.
<a href="#">as-path passes-through, on page 24</a>	Verifies if the supplied integer or parameter appears anywhere in the AS path or if the supplied sequence of integers and parameters appears, in the same order, anywhere in the AS path.
<a href="#">as-path unique-length, on page 29</a>	Performs specific checks based on the length of the AS path.

## as-path originates-from

To compare an AS path against the AS sequence beginning with the AS number that originated a route, use the **as-path originates-from** command in route-policy configuration mode.

**as-path originates-from** *as-number-list* [**exact**]

### Syntax Description

<b>as-number-list</b>	Numbers or parameters, enclosed in single quotation marks, that represent a sequence of autonomous system numbers. <ul style="list-style-type: none"> <li>• Range for 2-byte Autonomous system numbers (ASNs) is 1 to 65535.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asplain format is 1 to 4294967295.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asdot format is 1.0 to 65535.65535.</li> </ul>
<b>exact</b>	(Optional) Specifies that with the <b>exact</b> keyword, the <i>as-number-list</i> value must identically match the AS path for the route; without the <b>exact</b> keyword, any element in the <i>as-number-list</i> argument matches one or more occurrences of that element in the AS path for the route.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>as-number-list</i> 4-byte number range 1.0 to 65535.65535 was supported.
Release 3.9.0	Asplain format for 4-byte Autonomous system numbers notation was supported.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path originates-from** command as a conditional expression within an **if** statement to compare an AS path to the autonomous system sequence.

**Note**

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **originates-from** operator is similar to the **neighbor-is** operator, except that it looks at the autonomous system number at the opposite end of the AS path. In other words, it is comparing to the autonomous system that originated the route. It can take numbers or parameters, enclosed in single quotation marks, that represent a sequence of autonomous system numbers. When more than one number is specified in the list, the sequence of autonomous system numbers listed must appear as a subsequence in the AS path, with the last number corresponding to the autonomous system that originated the route.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following are incomplete configuration examples:

```
RP/0/0/CPU0:router(config-rpl)# if as-path originates-from '10 11' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path originates-from '$asnum 11' then
```

The first line of the preceding example evaluates to true if autonomous system 11 originated the route and then advertised it to autonomous system 10, from which the route was eventually propagated to us. In the case where the route has been aggregated, and the location of the originating autonomous system contains an AS-set, the **originates-from** operator evaluates to true if the argument to the **originates-from** operator is contained in the AS-set.

Without the **exact** keyword, repeated autonomous system numbers in the AS path are ignored. For example,

```
RP/0/0/CPU0:router(config-rpl)# if as-path originates-from '10 11' then
```

matches an autonomous system path ending

```
...10 10 10 11
```

and an autonomous system path ending

```
...10 11
```

With the **exact** keyword, repetitions are not ignored, therefore

```
RP/0/0/CPU0:router(config-rpl)# if as-path originates-from '10 11' exact then
```

matches the second of these autonomous system paths but not the first.

**Related Commands**

Command	Description
<a href="#">as-path in</a> , on page 12	Matches the AS path of a route to an AS path set.

Command	Description
<a href="#">as-path length, on page 16</a>	Compares the number of ASN in the AS path of a route
<a href="#">as-path passes-through, on page 24</a>	Verifies if the supplied integer or parameter appears anywhere in the AS path or if the supplied sequence of integers and parameters appears, in the same order
<a href="#">as-path unique-length, on page 29</a>	Performs specific checks based on the length of the AS path.

## as-path passes-through

To verify if the supplied integer or parameter appears anywhere in the AS path or if the supplied sequence of integers and parameters appears, in the same order, anywhere in the AS path, use the **as-path passes-through** command in route-policy configuration mode.

**as-path passes-through** *as-number-list* [**exact**]

### Syntax Description

<i>as-number-list</i>	Numbers or parameters, enclosed in single quotation marks, that represent a sequence of autonomous system numbers. <ul style="list-style-type: none"> <li>• Range for 2-byte Autonomous system numbers (ASNs) is 1 to 65535.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asplain format is 1 to 4294967295.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asdot format is 1.0 to 65535.65535.</li> </ul>
<b>exact</b>	(Optional) Specifies that with the <b>exact</b> keyword, the <i>as-number-list</i> value must identically match the AS path for the route; without the <b>exact</b> keyword, any element in the <i>as-number-list</i> argument matches one or more occurrences of that element in the AS path for the route.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>as-number-list</i> 4-byte number range 1.0 to 65535.65535 was supported.
Release 3.9.0	Asplain format for 4-byte Autonomous system numbers notation was supported.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.



Use the **as-path passes-through** command as a conditional expression within an **if** statement to verify if the specified integer or parameter appears anywhere in the AS path or if the sequence of integers and parameters appears.



**Note** For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **passes-through** operator takes a sequence of integers or parameters, enclosed in single quotation marks, as an argument. It can also take a single integer or parameter as an argument. It evaluates to true if the supplied integer or parameter appears anywhere in the AS path, or if the supplied sequence of integers and parameters appears, in the same order, anywhere in the AS path. This includes the **originates-from** or **neighbor-is** location in the AS path.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following are incomplete configuration examples:

```
RP/0/0/CPU0:router(config-rpl)# if as-path passes-through '10' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path passes-through '$asnum' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path passes-through '10 11' then
RP/0/0/CPU0:router(config-rpl-if)# if as-path passes-through '10 $asnum 12' then
```

Without the **exact** keyword, repeated autonomous system numbers in the AS path are ignored. For example:

```
RP/0/0/CPU0:router(config-rpl)# if as-path passes-through '9 10 11' then
```

matches an AS path containing

```
...9 10 10 10 11 ....
```

and an AS path containing:

```
...9 10 11...
```

With the **exact** keyword, repetitions are not ignored. Therefore:

```
RP/0/0/CPU0:router(config-rpl)# if as-path passes-through '9 10 11' exact then
```

matches the second of these AS paths but not the first.

## Related Commands

Command	Description
<a href="#">as-path in</a> , <a href="#">on page 12</a>	Matches the AS path of a route to an AS path set.
<a href="#">as-path length</a> , <a href="#">on page 16</a>	Compares the number of ASN in the AS path of a route

Command	Description
<a href="#">as-path originates-from, on page 21</a>	Compares an AS path to the AS sequence beginning with the AS number that originated a route.
<a href="#">as-path unique-length, on page 29</a>	Performs specific checks based on the length of the AS path.

## as-path-set

To create a named AS path set, use the **as-path-set** command in global configuration mode. To remove the named AS path set, use the **no** form of this command.

**as-path-set** *name*

**no as-path-set** *name*

### Syntax Description

<i>name</i>	Name of the AS path set.
-------------	--------------------------

### Command Default

No default behavior or values

### Command Modes

global configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path-set** command to create a named AS path set.

An AS path set comprises operations for matching an AS path attribute.

This command enters AS path set configuration mode, in which you can use the **ios-regex** keyword to indicate the type of regular expression. Single quotation marks are required around the regular expression.

The inline set form is a parenthesized list of comma-separated expressions.

See the “Understanding Regular Expressions, Special Characters and Patterns” appendix in the *Cisco IOS XR Getting Started Guide for the Cisco XR 12000 Series Router* for information about forming regular expressions.

### Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following is a sample definition of an AS path set named aset1:

```
RP/0/0/CPU0:router(config)# as-path-set aset1  
RP/0/0/CPU0:router(config-as)# ios-regex '_42$',  
RP/0/0/CPU0:router(config-as)# ios-regex '_127$'  
RP/0/0/CPU0:router(config-as)# end-set
```

This AS path set is composed of two elements. When used in a matching operation, this AS path set matches any route whose AS path ends with either the autonomous system number 42 or 127.

The following is a sample of an inline set:

```
RP/0/0/CPU0:router(config-rpl)# if as-path in (ios-regex '_42$', ios-regex$ '_127$')  
RP/0/0/CPU0:router(config-rpl-if)# pass  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

## as-path unique-length

To perform specific checks based on the length of the AS path (match against the number of unique ASNs in the AS path), use the **as-path unique-length** command in route-policy configuration mode.

**as-path unique-length** {**eq**|**is**|**ge**|**le**} {*number*|*parameter*}

### Syntax Description

<b>eq</b>   <b>is</b>   <b>ge</b>   <b>le</b>	Equal to; greater than or equal to; less than or equal to.
<i>number</i>	Value assigned to an 11-bit unsigned integer. Range is from 0 to 2047.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **as-path unique-length** command as a conditional expression within an **if** statement to perform a match based on the length of the AS path.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **unique-length** operator is similar to the length operator, except that when an AS path has been padded with the same autonomous system number multiple times, the operator counts only one when the route is padded. Therefore, given an AS path of 333 333 111 222 123 444 444 444, the **unique-length** operator would return a value of 5, whereas the length operator would return a value of 8.

### Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows how to perform checks based on the AS path length. If the AS path matches the specified values, the local preference is set to 100:

```
RP/0/0/CPU0:router(config-rpl)# if as-path unique-length eq 10 then
RP/0/0/CPU0:router(config-rpl-if)# if as-path unique-length ge 10 then
RP/0/0/CPU0:router(config-rpl-if)# if as-path unique-length le 10 then

RP/0/0/CPU0:router(config-rpl)# if as-path unique-length eq $integerparam then
RP/0/0/CPU0:router(config-rpl-if)# if as-path unique-length ge $geparam then
RP/0/0/CPU0:router(config-rpl-if)# if as-path unique-length le $leparam then

RP/0/0/CPU0:router(config-rpl)# set local-preference 100
RP/0/0/CPU0:router(config-rpl)# endif
```

## Related Commands

Command	Description
<a href="#">as-path length</a> , on page 16	Performs conditional checks based on the length of the AS path.

# community is-empty

To check if a route has no community attributes associated with it, use the **community is-empty** command in route-policy configuration mode.

**community is-empty**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Command History	Release	Modification
	Release 3.2	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **community is-empty** command as a conditional expression within an **if** statement to check if a route has community attributes associated with it.



**Note**

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command takes no arguments and evaluates to true only if the route has no community attributes associated with it.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** In the following example, if the route has no community attributes associated with it, then the local preference is set to 100:

```
RP/0/0/CPU0:router(config-rpl)# if community is-empty then
RP/0/0/CPU0:router(config-rpl-if) # set local-preference 100
RP/0/0/CPU0:router(config-rpl-if) # endif
```

## community matches-any

To match any elements of a community set, use the **community matches-any** command in route-policy configuration mode.

**community matches-any** {*community-set-name*| *inline-community-set*| *parameter*}

### Syntax Description

<i>community-set-name</i>	Name of a community set.
<i>inline-community-set</i>	Inline community set. The inline community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.  The BGP community matching operators were enhanced to allow comparisons on ranges and regular expressions.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **community matches-any** command as a conditional expression within an **if** statement to match any element of a community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-any** operator evaluates as true if at least one community element of the community attribute for the route matches an element in the community set operand. If no community in the route matches any of the specifications in the named or inline set, then the condition evaluates to false. Likewise, when there is no community at all in the route, the condition evaluates to false.



Matching of a community in the route to a specification in a named or an inline set is intuitive. If the community specification in a set is the familiar colon-separated decimal 16-bit numbers specification, or one of the well-known communities, the community matches the specification if the specification denotes the same 32-bit number as that in the route. If the community specification uses a wildcard, then the community in the route matches if it is one of the many communities denoted by the wildcard specification. In inline sets, community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

Communities may also be matched using range and regular expression operators. Range specifications are entered as follows: [ *low-value* .. *high-value* ]. Either or both colon-separated halves of a community value may contain a range. The following are valid range specifications:

```
10:[100..1000]
[10..100]:80
[10..100]:[100..2000]
```

In addition, the **private-as** keyword may be used to specify the range from 64512 to 65534. Regular expressions are specified as the **ios-regex** keyword followed by a valid regular expression string.

Community values from the route are matched one at a time to the match specifications. Therefore, regex match specifications are expected to represent one individual community value and not a sequence of community values.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, a named community set called my-community-set and a route policy called community-matches-any-example are created. The policy sets the local-preference to 100 for any route that has one or more of the communities in the my-community-set community set. If the route does not have any of these communities, the policy checks whether it has any communities whose first half is in the range from 10 to 25 and whose second half is the value 35, in which case it sets the local-preference to 200. Otherwise, it checks for a community value in the range of 30:100 to 30:500, in which case it sets the local-preference to 300.

```
RP/0/0/CPU0:router(config)# community-set my-community-set
RP/0/0/CPU0:router(config-comm)# 10:20,
RP/0/0/CPU0:router(config-comm)# 10:30,
RP/0/0/CPU0:router(config-comm)# 10:40
RP/0/0/CPU0:router(config-comm)# end-set

RP/0/0/CPU0:router(config)# route-policy community-matches-any-example
RP/0/0/CPU0:router(config-rpl)# if community matches-any my-community-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif community matches-any ([10..25]:35) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-any (30:[100..500]) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

**Related Commands**

Command	Description
<a href="#">community matches-every, on page 35</a>	Matches every element of a community set.

## community matches-every

To match every element of a community set, use the **community matches-every** command in route-policy configuration mode.

**community matches-every** {*community-set-name*| *inline-community-set*| *parameter*}

### Syntax Description

<i>community-set-name</i>	Name of a community set.
<i>inline-community-set</i>	Inline community set. The inline community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.  The BGP community matching operators were enhanced to allow comparisons on ranges and regular expressions.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **community matches-every** command as a conditional expression within an **if** statement to match every element of a community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-every** operator evaluates as true if every specification in the named set or inline set specified matches at least one community value in the route. If any community specification in the named or inline set is not matched, then the operation evaluates to false.

Matching of a community in the route to a specification in a named or an inline set is intuitive. If the community-specification in a set is the familiar colon-separated decimal 16-bit numbers specification, or one

of the well-known communities, the community matches the specification if the specification denotes the same 32-bit number as that in the route. If the community specification uses a wildcard, then the community in the route matches if it is one of the many communities denoted by the wildcard specification. In inline sets, community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

Communities may also be matched using range and regular expression operators. Range specifications are entered as follows: [ *low-value* .. *high-value* ]. Either or both colon-separated halves of a community value may contain a range. The following are valid range specifications:

```
10:[100..1000]
[10..100]:80
[10..100]:[100..2000]
```

Therefore, a **matches-every** operation with two community range specifications means that a community must be present in the route that corresponds to each range. For example, in the following statement:

```
if community matches-every (10:[100..200],20:[100..200]) then
```

the statement evaluates as true if one or more communities in the route lie in the range 10:[100..200] and one or more communities in the route lie in the range 20:[100..200].

In addition, the **private-as** keyword may be used to specify the range from 64512 to 65534.

Regular expressions are specified as the **ios-regex** keyword followed by a valid single-quoted regular expression string. Community values from the route are matched one at a time against the match specifications. Therefore, regex match specifications are expected to represent one individual community value and not a sequence of community values.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the route policy named community-matches-every-example sets the local-preference value to 100 for all routes that have all three communities in the my-community-set community set. Routes that do not have all three communities but have a community that matches the first regular expression match have the local-preference value set to 200. Finally, any remaining routes that match the last regular expression have the local-preference values set to 300.

```
RP/0/0/CPU0:router(config)# community-set my-community-set
RP/0/0/CPU0:router(config-comm)# 10:20,
RP/0/0/CPU0:router(config-comm)# 10:30,
RP/0/0/CPU0:router(config-comm)# 10:40
RP/0/0/CPU0:router(config-comm)# end-set

RP/0/0/CPU0:router(config)# route-policy community-matches-every-example
RP/0/0/CPU0:router(config-rpl)# if community matches-every my-community-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-every (ios-regex
'10:[0-9]0_') then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# elseif community matches-every
(ios-regex'20:[0-9]0_') then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
```

```
RP/0/0/CPU0:router(config-rpl-elseif)# endif  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

**Related Commands**

Command	Description
<a href="#">community matches-any, on page 32</a>	Matches any element of a community set.

# community-set

To define a community set, use the **community-set** command in global configuration mode. To remove the community set, use the **no** form of this command.

**community-set** *name*

**no community-set** *name*

## Syntax Description

<i>name</i>	Name of the community set.
-------------	----------------------------

## Command Default

No default behavior or values

## Command Modes

global configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Regular expressions and ranges can be specified to match the communities. An attempt to use a community set that contains a range or regular expression to set a community value is rejected when an attempt to attach such a policy is made.

A community set holds community values for matching against the Border Gateway Protocol (BGP) community attribute. A community is a 32-bit quantity. For notational convenience, each community value must be split in half and expressed as two unsigned decimal integers in the range from 0 to 65535, separated by a colon.

The inline form of a community set also supports parameterization. Each 16-bit portion of the community may be parameterized.

The routing policy language (RPL) provides symbolic names for the standard well-known community values: **accept-own** is 0xFFFF0001, **internet** is 0:0, **no-export** is 65535:65281, **no-advertise** is 65535:65282, and **local-as** is 65535:65283.

RPL also provides a facility for using wildcards in community specifications. A wildcard is specified by inserting an asterisk (\*) in place of one of the 16-bit portions of the community specification, which indicates that any value for that portion of the community matches.

Every community set must contain at least one community value. An empty community set is invalid and the policy configuration system rejects it.

Community sets can be entered in these formats:

Format	Description
<i>#-remark</i>	Remark beginning with '#'
*	Wildcard (any community or part thereof)
<i>0-65535</i>	16-bit half-community number
[	Left bracket to begin range
<b>accept-own</b>	Accept-Own (BGP well-known community)
<b>dfa-regex</b>	DFA (deterministic finite automata) style regular expression
<b>internet</b>	Internet (BGP well-known community)
<b>ios-regex</b>	Traditional IOS style regular expression
<b>local-AS</b>	Do not send outside local AS (BGP well-known community)
<b>no-advertise</b>	Do not advertise to any peer (BGP well-known community)
<b>no-export</b>	Do not export to next AS (BGP well-known community)
<b>private-as</b>	Match within BGP private AS range [64512..65534]

**Note**

The dfa-regex and ios-regex syntax for community set is `"[/[^: & < > ]*/[^: & < > ]*/"`. This means that regex starts with a single-quote (") followed by a string of any character (that does not include single-quote, colon, ampersand, less-than, greater-than, or space) followed by a colon, and a string of any characters (that does not include single-quote, colon, ampersand, less-than, greater-than, or space) followed by single-quote.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, a community set named cset\_accept\_own is created:

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#community-set cset_accept_own
RP/0/0/CPU0:router(config-comm)#accept-own
RP/0/0/CPU0:router(config-comm)#end-set
```

In the following example, a community set named cset1 is created:

```
RP/0/0/CPU0:router(config)# community-set cset1
```

```
RP/0/0/CPU0:router(config-comm)# 12:34,  
RP/0/0/CPU0:router(config-comm)# 12:56,  
RP/0/0/CPU0:router(config-comm)# 12:78,  
RP/0/0/CPU0:router(config-comm)# internet  
RP/0/0/CPU0:router(config-comm)# end-set
```

In the following example, a community set named cset2 is created:

```
RP/0/0/CPU0:router(config)# community-set cset2  
RP/0/0/CPU0:router(config-comm)# 123:456,  
RP/0/0/CPU0:router(config-comm)# no-advertise,  
RP/0/0/CPU0:router(config-comm)# end-set
```

In the following example, a community set named cset3 is created. This policy uses wildcards and matches all communities where the autonomous system part of the community is 123.

```
RP/0/0/CPU0:router(config)# community-set cset3  
RP/0/0/CPU0:router(config-comm)# 123:*  
RP/0/0/CPU0:router(config-comm)# end-set
```



# delete community

To delete community attributes associated with a Border Gateway Protocol (BGP) route, use the **delete community** command in route-policy configuration mode.

**delete community** {**all** | **in** {*community-set-name* | *inline-community-set* | *parameter*} | **not in** {*community-set-name* | *inline-community-set* | *parameter*}}

## Syntax Description

<b>all</b>	Removes all communities except the well-known communities.
<b>in</b>	Removes any communities associated with the route that are listed in either the named community set or the inline community set.
<i>community-set-name</i>	Name of a community set.
<i>inline-community-set</i>	Inline community set. The inline community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>not in</b>	Removes all communities that are not listed in either the named community set or the inline community set, and are not well-known communities.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.  The BGP community deletion operations were extended to allow communities that match a range specification or a regular expression to be deleted.
Release 3.4.0	The <i>parameter</i> argument was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **delete community** command to delete community attributes associated with a BGP route.

**Note**

The **delete community** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Communities are 32-bit values carried in Border Gateway Protocol (BGP) routes. Each route may have zero or more communities in an unordered list.

You can remove a well-known community (internet, no-export, no-advertise, or local-as) from a route, but this removal must be done explicitly. This command should be used with a degree of caution. In general, few circumstances exist in which you would need to remove a well-known community.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following example shows how to delete any communities associated with the routes that are listed in either the named community set or inline community set, respectively.

```
RP/0/0/CPU0:router(config-rpl)# delete community in my_community_set
RP/0/0/CPU0:router(config-rpl)# delete community in (10:[0..50],20:[60..80])
```

The following example shows how to remove all communities including well-known communities.

```
RP/0/0/CPU0:router(config-rpl)# delete community in (internet, no-export, no-advertise,
local-as, *:*)
```

The following example shows how to remove all communities except for the well-known communities.

```
RP/0/0/CPU0:router(config-rpl)# delete community all
```

The following example shows how to delete the well-known community value internet from a route:

```
RP/0/0/CPU0:router(config-rpl)# delete community in (internet)
```

# delete extcommunity rt

To delete route target (RT) extended community attributes associated with a Border Gateway Protocol (route), use the **delete extcommunity rt** command in route-policy configuration mode.

**delete extcommunity rt** {**all**| **in** {*extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}| **not in** {*extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}}

## Syntax Description

<b>all</b>	Removes all extended communities.
<b>in</b>	Removes any extended communities associated with the routes that are listed in either the named extended community set or the inline extended community set.
<i>extcommunity-set-name</i>	Name of an extended community set.
<i>inline-extcommunity-set</i>	Inline extended community set. The inline extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>not in</b>	Removes all extended communities that are not listed in either the named extended community set or the inline extended community set, and are not well-known extended communities.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.
Release 3.5.0	Support was added for the usage of Wildcards (*) and regular expressions for extended community set elements.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **delete extcommunity rt** command to delete extended community values from a BGP route target extended community list in a route.

**Note**

The **delete extcommunity rt** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Extended communities are similar to regular Border Gateway Protocol (BGP) communities but contain more data and have a richer structure for encoding information in them.

Extended communities can be in the following forms: SoO:AS:tag, SoO:IP:tag, RT:AS:tag, or RT:IP:tag.

Wildcards (\*) and regular expressions are allowed for extended community set elements.

The forms of this command that take a named extended community set or an inline extended community set value as arguments are equivalent. They delete any extended communities that are listed in either the named set or the inline set, respectively.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, all extended communities are deleted:

```
RP/0/0/CPU0:router(config-rpl)# delete extcommunity rt all
```

In this example, any extended communities that are listed in my-extcommunity-set are deleted:

```
RP/0/0/CPU0:router(config-rpl)# delete extcommunity rt in my-extcommunity-set
```

In this example, extended communities associated with the route listed in the named inline extended community sets are deleted:

```
RP/0/0/CPU0:router(config-rpl)# delete extcommunity rt in (67:29, 67:55)
```

## destination in

To match a destination entry in a named prefix set or inline prefix set, use the **destination in** command in route-policy configuration mode.

**destination in** {*prefix-set-name*| *inline-prefix-set*| *parameter*}

### Syntax Description

<i>prefix-set-name</i>	Name of a prefix set.
<i>inline-prefix-set</i>	Inline prefix set. The inline prefix set must be enclosed in parentheses.
<i>parameter</i> parameter	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **destination in** command as a conditional expression within an **if** statement to match a destination entry in a named prefix set or inline prefix set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command takes either a named prefix set or an inline prefix set value as an argument. The condition returns true if the destination entry matches any entry in the prefix set or inline prefix set. An attempt to match a destination using a prefix set that is defined but contains no elements returns false.

The routing policy language (RPL) provides the ability to test destinations for a match to a list of prefix match specifications using the **in** operator. The **destination in** command is protocol-independent.

In Border Gateway Protocol (BGP), the destination of a route is also known as its network-layer reachability information (NLRI). It comprises a prefix value and a mask length.

RPL supports both 32-bit IPv4 prefixes, specified in dotted-decimal format, and 128-bit IPv6 prefixes, specified in colon-separated hexadecimal format.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, a prefix set named my-prefix-set is defined and a route policy named use-destination-in is created. Within the use-destination-in route policy, the **destination in** command is used within an **if** statement to learn if the destination is in the prefix-set named my-prefix-set. If it is, then local preference is set to 100. If it is not in my-prefix-set but does match the next prefix specifications, then local preference is set to 200.

```
RP/0/0/CPU0:router(config)# prefix-set my-prefix-set
RP/0/0/CPU0:router(config-pfx)# 10.0.0.1/32,
RP/0/0/CPU0:router(config-pfx)# fe80::203:0:0:0/64,
RP/0/0/CPU0:router(config-pfx)# 10.0.0.2/24 le 32
RP/0/0/CPU0:router(config-pfx)# end-set

RP/0/0/CPU0:router(config)# route-policy use-destination-in
RP/0/0/CPU0:router(config-rpl)# if destination in my-prefix-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif destination in (10.0.0.1/32, 10.0.0.2/24 le 32)
then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

In the following example, a prefix set named ipv6-prefix-set is defined and a route policy named ipv6-destination-in is created. Within the ipv6-destination-in route policy, the **destination in** command is used within an **if** statement to learn if the destination is in the prefix-set named ipv6-prefix-set. If it is, then the next-hop is set to 2001:abcd:fedc::1. If it is not in ipv6-prefix-set but does match the next prefix specifications, then the next-hop is set to 1111:2222:3333:4444:5555:6666:7777:8888.

```
RP/0/0/CPU0:router(config)# prefix-set ipv6-prefix-set
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:1::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:2::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:3::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:4::/64
RP/0/0/CPU0:router(config-pfx)# end-set

RP/0/0/CPU0:router(config)# route-policy ipv6-destination-in
RP/0/0/CPU0:router(config-rpl)# if destination in ipv6-prefix-set then
RP/0/0/CPU0:router(config-rpl-if)# set next-hop 2001:abcd:fedc::1
RP/0/0/CPU0:router(config-rpl-if)# elseif destination in (2001::1, 2002:1:2:3::/64) then
RP/0/0/CPU0:router(config-rpl-elseif)# set next-hop 1111:2222:3333:4444:5555:6666:7777:8888

RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# done

To stop executing a policy and accept the route, use the **done** command in route-policy configuration mode.

**done**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Release	Modification
Release 3.5.0	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **done** command to stop executing the policy and accept the route.



**Note** The **done** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

When encountering a **done** statement the route is passed and no further policy statements are executed. All modifications made to the route prior to the **done** statement are still valid.



**Note** The default action of a route policy is to drop or discard any routes that have not been either explicitly passed or for which no attempt has been made to modify with an action. The routing policy language (RPL) does not have specific “match clauses,” which means the default drop behavior is controlled by whether a route has been explicitly passed or an attempt has been to modify the route using an action statement.

Task ID	Operations
route-policy	read, write

## Examples

In the following example, if the destination match succeeds for 29.0.0.0/8 le 32, the execution continues past `set community 102:12` and onto the next statement. If the destination match succeeds for 39.0.0.0/8 le 32 execution, then the policy execution stops when it encounters the *done* statement.

```
RP/0/0/CPU0:router(config)# route-policy done_st_example
RP/0/0/CPU0:router(config-rpl)# if destination in (29.0.0.0/8 le 32) then
RP/0/0/CPU0:router(config-rpl-if)# set community 102:12
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# if destination in (39.0.0.0/8 le 32) then
RP/0/0/CPU0:router(config-rpl-if)# set community 102:39
RP/0/0/CPU0:router(config-rpl-if)# done
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# if destination in (49.0.0.0/8 le 32) then
RP/0/0/CPU0:router(config-rpl-if)# set community 102:49
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# if destination in (59.0.0.0/8 le 32) then
RP/0/0/CPU0:router(config-rpl-if)# set community 102:59
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```



# drop

To discard a route, use the **drop** command in route-policy configuration mode.

**drop**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Release	Modification
Release 3.2	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **drop** command within a route policy to drop a route.



**Note**

The **drop** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

This command causes the route to be dropped. After a route is dropped, no further execution of policy occurs. Therefore, if after executing the first two statements of a policy the **drop** statement is encountered, the route is discarded and execution stops immediately even when the policy contains further statements.



**Note**

The default action of a route policy is to drop or discard any routes that have not been either explicitly passed or attempted to be modified with an action. The routing policy language (RPL) does not have specific “match clauses,” which means the default drop behavior is controlled by whether a route has been explicitly passed or an attempt has been to modify the route using an action statement.

**Task ID**

Task ID	Operations
route-policy	read, write

## Examples

In the following example, any route with a destination address contained within the prefix set pset1 is dropped:

```
RP/0/0/CPU0:router(config-rpl)# if destination in pset1 then  
RP/0/0/CPU0:router(config-rpl-if)# drop  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

# edit

To edit the contents of a route policy, a prefix set, an AS path set, a community set, or an extended community set, use the **edit** command in EXEC configuration mode.

**edit** {**route-policy**|**prefix-set**|**as-path-set**|**community-set**|**extcommunity-set** {**rt**|**soo**}|**policy-global**|**rd-set**} *name* [**nano**|**emacs**|**vim**|**inline** {**add**|**prepend**|**remove**} *set-element*]

## Syntax Description

<b>route-policy</b>	Edits the contents of a route policy.
<b>prefix-set</b>	Edits the contents of a prefix set.
<b>as-path-set</b>	Edits the contents of an AS path set.
<b>community-set</b>	Edits the contents of a community set.
<b>extcommunity-set</b>	Edits the contents of an extended community set of the specified type.
<b>rt</b>	Edits the BGP route target (RT) extended community.
<b>soo</b>	Edits the BGP site of origin (SoS) extended community.
<b>policy-global</b>	Edits the contents of policy-global definitions.
<b>rd-set</b>	Edits the contents of a route-distinguisher set.
<i>name</i>	Name of a route policy, a prefix set, an AS path set, a community set, or an extended community set, RD set, or global parameters.
<b>nano</b>	(Optional) Uses GNU Nano text editor.
<b>emacs</b>	(Optional) Uses Micro Emacs editor.
<b>vim</b>	(Optional) Uses VI Improved editor.
<b>inline</b>	(Optional) Uses the command line.
<b>add</b>	Appends the element to the set.
<b>prepend</b>	Prepends the element to the set.
<b>remove</b>	Removes the element from the set.
<i>set-element</i>	Value of the set element.
<b>Note</b> To inline edit multiple set elements separated with comma, use quotes to club the entries as a single argument. Example: edit extcommunity-set rt rt_set inline add "4:4,5:4"	

**Command Default** Default editor is GNU nano text editor

**Command Modes** EXEC configuration

Command History	Release	Modification
	Release 3.2	This command was introduced.
	Release 3.3.0	Added <b>vim</b> keyword.
	Release 3.4.0	The <b>policy-global</b> keyword was added.
	Release 3.5.0	The <b>nano</b> keyword was added and GNU nano text editor became the default editor type.
	Release 3.6.0	The following keywords and arguments were added: <ul style="list-style-type: none"> <li>• <b>inline</b></li> <li>• <b>add</b></li> <li>• <b>prepend</b></li> <li>• <b>remove</b></li> <li>• <i>set-element</i></li> </ul>

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **edit** command to edit the contents of a route policy, a prefix set, an AS path set, a community set, an extended community set, a global policy, or a route destination set.

After editing with Nano, save the edit buffer and exit the editor using the Ctrl-X keystroke.

After editing with Emacs, save the editor buffer by using the Ctrl-X and Ctrl-S keystrokes. To save and exit the editor, use the Ctrl-X and Ctrl-C keystrokes.

After editing with VIM, to write to a current file and exit use the :wq or :x or ZZ keystrokes. To quit and confirm, use the :q keystrokes. To quit and discard changes, use the :q! keystrokes.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples**

In the following example, the policy\_A policy is opened in the editor:

```
RP/0/0/CPU0:router# edit route-policy policy_A
```

```
-----
== MicroEMACS 3.8b () == rpl_edit.139281 ==
  if destination in (2001::/8) then
    drop
  endif
end-policy
!

== MicroEMACS 3.8b () == rpl_edit.139281 ==
Parsing.
83 bytes parsed in 1 sec (82)bytes/sec
Committing.
1 items committed in 1 sec (0)items/sec
Updating.
Updated Commit database in 1 sec
```

If there are parse errors, you are asked whether editing should continue:

```
RP/0/0/CPU0:router#edit route-policy policy_B
== MicroEMACS 3.8b () == rpl_edit.141738
route-policy policy_B
  set metric-type type_1
  if destination in (2001::/8) then
    drop
  endif
end-policy
!
== MicroEMACS 3.8b () == rpl_edit.141738 ==
Parsing.
105 bytes parsed in 1 sec (103)bytes/sec

% Syntax/Authorization errors in one or more commands.!! CONFIGURATION
FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
  set metric-type type_1
  if destination in (2001::/8) then
    drop
  endif
end-policy
!

Continue editing? [no]:
```

If you answer **yes**, the editor continues on the text buffer from where you left off. If you answer **no**, the running configuration is not changed and the editing session is ended.

After the policy is opened, it may be manipulated using normal editor commands, then saved and committed to the running configuration.

# end-global

To end the definition of global parameters and exit global parameter configuration mode, use the **end-global** command in global parameter configuration mode.

**end-global**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Global parameter configuration

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **end-global** command to end the definition of global parameters and exit global parameter configuration mode.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the **end-global** command ends the definition of global parameters:

```
RP/0/0/CPU0:router(config)#policy-global
RP/0/0/CPU0:router(config-rp-gl)# glbpathtype 'ebgp'
RP/0/0/CPU0:router(config-rp-gl)# glbtag '100'
RP/0/0/CPU0:router(config-rp-gl)# end-global
```

## Related Commands

Command	Description
<a href="#">policy-global</a> , on page 102	Enters global parameter configuration mode.

# end-policy

To end the definition of a route policy and exit route-policy configuration mode, use the **end-policy** command in route-policy configuration mode.

**end-policy**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Command History	Release	Modification
	Release 3.2	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **end-policy** command to end the definition of a route policy and exit route-policy configuration mode.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** In the following example, the **end-policy** command ends the definition of a route policy:

```
RP/0/0/CPU0:router(config)#route-policy med-to-local-pref
RP/0/0/CPU0:router(config-rpl)#if med eq 150 then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 10
RP/0/0/CPU0:router(config-rpl-if)# elseif med eq 200 then
RP/0/0/CPU0:router(config-elseif)# set local-preference 60
RP/0/0/CPU0:router(config-elseif)# elseif med eq 250 then
RP/0/0/CPU0:router(config-elseif)# set local-preference 0

RP/0/0/CPU0:router(config-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

**Related Commands**

Command	Description
<a href="#">route-policy (RPL), on page 120</a>	Defines a route policy and enters route-policy configuration mode.



# end-set

To end the definition of an AS path set, a prefix set, a community set, an extended community set, or an RD set and return to global configuration mode, use the **end-set** command in route-policy configuration mode.

**end-set**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

AS path set configuration  
Prefix set configuration  
Community set configuration  
Extended community set configuration  
Route distinguisher set configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	Added support functionality for the <b>rd-set</b> command.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **end-set** command to end the definition of an AS path set, a prefix set, a community set, or an extended community set.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the **end-set** command ends the definition of an AS path set named aset1:

```
RP/0/0/CPU0:router(config)# as-path-set aset1
RP/0/0/CPU0:router(config-as)# ios-regex '_42$',
RP/0/0/CPU0:router(config-as)# ios-regex '_127$'
```

```
RP/0/0/CPU0:router(config-as)# end-set
RP/0/0/CPU0:router(config)#
```

The following example shows how to create an RD set called `my_rd_set` and use the `end-set` command to end the definition:

```
RP/0/0/CPU0:router(config)# rd-set my_rd_set
RP/0/0/CPU0:router(config-rd)# 172.16.0.0/16:*,
RP/0/0/CPU0:router(config-rd)# 172.17.0.0/16:100,
RP/0/0/CPU0:router(config-rd)# 192:*,
RP/0/0/CPU0:router(config-rd)# 192:100
RP/0/0/CPU0:router(config-rd)# end-set
```

# extcommunity rt is-empty

To check if a Border Gateway Protocol (BGP) route has route target (RT) extended community attributes associated with it, use the **extcommunity rt is-empty** command in route-policy configuration mode.

**extcommunity rt is-empty**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or value

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity is-empty</b> command.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity rt is-empty** command as a conditional expression within an **if** statement to check if a BGP route has extended community attributes associated with it.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **is-empty** operator takes no arguments and evaluates to true if the route has no extended community attributes associated with it.


## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, if the extended community is empty, then the local preference is set to 100:

```
RP/0/0/CPU0:router(config)# route-policy extcommunity-is-empty-example
RP/0/0/CPU0:router(config-rpl)# if extcommunity rt is-empty then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
```

 `extcommunity rt is-empty`

```
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## extcommunity rt matches-any

To match any element of a Border Gateway Protocol (BGP) route target (RT) extended community set, use the **extcommunity rt matches-any** command in route-policy configuration mode.

**extcommunity rt matches-any** {*extcommunity-set-name*|*inline-extcommunity-set*|*parameter*}

### Syntax Description

<i>extcommunity-set-name</i>	Name of an RT extended community set.
<i>inline-extcommunity-set</i>	Inline RT extended community set. The inline extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity matches-any</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.
Release 3.5.0	Wildcards (*) and regular expressions are allowed for extended community set elements.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity rt matches-any** command as a conditional expression within an **if** statement to match elements of an extended community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-any** operator evaluates as true if at least one extended community in the route matches an extended community specification in the named or inline set. If no extended community

in the route matches any of the specifications in the named or inline set, then this simple condition evaluates to false. Likewise, when there is no extended community at all in the route, the condition evaluates to false.

Matching an extended community in the route to a specification in a named or an inline set is intuitive. In inline sets, extended community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, an extended community set named my-extcommunity-set and a parameterized route-policy named my-extcommunity-set-example(\$tag,\$ip) are defined. The **extcommunity rt matches-any** command is used in an if statement such that if at least one extended community in the route matches an extended community specification in the named set, then the local preference is set to 100. If there is no extended community in the route that matches any of the specifications in the named set, then the condition evaluates as false and the extended community is compared to the inline extended sets.

```
RP/0/0/CPU0:router(config)# extcommunity-set rt my-extcommunity-set
RP/0/0/CPU0:router(config-ext)# 10:615,
RP/0/0/CPU0:router(config-ext)# 10:6150,
RP/0/0/CPU0:router(config-ext)# 15.15.15.15:15
RP/0/0/CPU0:router(config-ext)# end-set

RP/0/0/CPU0:router(config)# route-policy my-extcommunity-set-example($tag,$ip)
RP/0/0/CPU0:router(config-rpl)# if extcommunity rt matches-any my-extcommunity-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif extcommunity rt matches-any (10:20, 10:$tag) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# elseif extcommunity rt matches-any ($ip:$tag) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/0/CPU0:router(config-rpl-elseif)# elseif extcommunity rt matches-any (2.3.4.5:$tag)
then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 400
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">extcommunity rt matches-every, on page 63</a>	Matches every element of a BGP RT extended community set.

## extcommunity rt matches-every

To match every element of a Border Gateway Protocol (BGP) route target (RT) extended community set, use the **extcommunity rt matches-every** command in route-policy configuration mode.

**extcommunity rt matches-every** {*extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}

### Syntax Description

<i>extcommunity-set-name</i>	Name of an RT extended community set.
<i>inline-extcommunity-set</i>	Inline RT extended community set. The inline extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity matches-every</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.
Release 3.5.0	Support was added for the usage of Wildcards (*) and regular expressions for extended community set elements.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity rt matches-every** command as a conditional expression within an **if** statement to match every element of an RT extended community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-every** operator evaluates as true if every extended community value in the extended community attribute for the route matches at least one element of the extended community set or inline set. If no extended community in the route matches any of the specifications in the named or

inline set, then this simple condition evaluates to false. Likewise, when there is no extended community at all in the route, the condition evaluates to false.

Matching an extended community in the route to a specification in a named or an inline set is intuitive. In inline sets, extended community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, an extended community set named my-extcommunity-set and a parameterized route policy named extcommunity-matches-every-example (\$as, \$tag) are defined. The condition extcommunity rt matches-every is used in an if statement in this policy. If it evaluates to true, the local-preference value is set to 100. If it evaluates to false, the extended community is evaluated using an inline set. If that condition evaluates to true, the local-preference value is set to 200. If it evaluates to false, the local-preference value is set to 300.

```
RP/0/0/CPU0:router(config)# extcommunity-set rt my-extcommunity-set
RP/0/0/CPU0:router(config-ext)# 10:20,
RP/0/0/CPU0:router(config-ext)# 10:30,
RP/0/0/CPU0:router(config-ext)# 10:40
RP/0/0/CPU0:router(config-ext)# end-set
RP/0/0/CPU0:router(config)# route-policy extcommunity-matches-every-example($as,$tag)
RP/0/0/CPU0:router(config-rpl)# if extcommunity rt matches-every my-extcommunity-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif extcommunity rt matches-every (10:20, 10:$tag,
$as:30) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# elseif
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">extcommunity rt matches-any, on page 61</a>	Matches any element of a BGP RT extended community set.



## extcommunity rt matches-within

To match at least one element of an extended community set of a Border Gateway Protocol (BGP) route target (RT), use the **extcommunity rt matches-within** command in route-policy configuration mode.

**extcommunity rt matches-within** {*rt-type-extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}

### Syntax Description

<i>rt-type-extcommunity-set-name</i>	Name of an RT extended community set.
<i>inline-extcommunity-set</i>	Inline RT extended community set, enclosed in parentheses.
<i>parameter</i>	Parameter name preceded with a "\$" symbol.

### Command Default

None

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 4.2.0	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity rt matches-within** command as a conditional expression within an if statement to match elements of an extended community set.



#### Note

For a list of all conditional expressions available within an if statement, see the if command.

A simple condition using the matches-within operator evaluates as true if all the elements in extended community from the route match any element in the extended community set. For example, let 'c' be the RTs from the route and 'm' be the RT set from the policy. With the **extcommunity rt matches-within** configuration, each value in 'c' must match any (or at least one) value in 'm'.

Matching an extended community in the route to a specification in a named or an inline set is intuitive. In inline sets, extended community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

**Task ID**

Task ID	Operation
route-policy	read, write

**Examples**

In the following example, an extended community set named *my-extcommunity-set* and a parameterized route-policy named *my-extcommunity-set-example(\$tag,\$ip)* are defined. The **extcommunity rt matches-within** command is used in an if statement such that if all extended community values in the route match any element of the extended community specification in the named set, then the local preference is set to 100.

```
RP/0/0/CPU0:router(config)#extcommunity-set rt my-extcommunity-set
RP/0/0/CPU0:router(config-ext)#10:615,
RP/0/0/CPU0:router(config-ext)#10:6150,
RP/0/0/CPU0:router(config-ext)#15.15.15.15:15
RP/0/0/CPU0:router(config-ext)#end-set
RP/0/0/CPU0:router(config)#route-policy my-extcommunity-set-example($tag,$ip)
RP/0/0/CPU0:router(config-rpl)#if extcommunity rt matches-within my-extcommunity-set then
RP/0/0/CPU0:router(config-rpl-if)#set local-preference 100
```

# extcommunity-set cost

To define a cost extended community set, use the **extcommunity-set cost** command in global configuration mode. To remove the cost extended community set, use the **no** form of this command.

**extcommunity-set cost** *name*

**no extcommunity-set cost** *name*

## Syntax Description

<i>name</i>	Name of a cost extended community set. The <i>name</i> argument is case sensitive, can contain any alphanumeric characters, and can be up to 63 characters in length.
-------------	---

## Command Default

No default behavior or values

## Command Modes

global configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
Release 3.5.0	Support was added for the usage of Wildcards (*) and regular expressions for extended community set elements.
Release 3.9.0	Support was added for more cost extended community formats.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity-set cost** command to define a cost extended community set.

An extended community set is analogous to a community set except that it contains extended community values instead of regular community values. Extended community values are 64-bit structured values. An extended community set also supports named forms and inline forms.

Cost extended communities can be entered in these formats:

- **#-remark** ---Remark beginning with '#'
- **0-255**---Decimal number
- **abort** ---Discard RPL definition and return to top level config
- **end-set** ---End of set definition
- **exit** ---Exit from the submode

- **igp**:---Cost Community with IGP as point of insertion
- **pre-bestpath**: ---Cost Community with Pre-Bestpath as point of insertion
- **show** ---Show partial RPL configuration

Multiple cost community set clauses can be configured in each route policy block or sequence. Each cost community set clause must have a different ID (0-255). The cost community set clause with the lowest cost-value is preferred by the best path selection process when all other attributes are equal.

As with community sets, the inline form supports parameterization within parameterized policies. Either portion of the extended community value can be parameterized.

Every extended community set must contain at least one extended community value. Empty extended community sets are invalid and the policy configuration system rejects them.

Wildcards (\*) and regular expressions are allowed for extended community set elements.

## Examples

In the following example, a cost extended community set named extcomm-cost is defined:

```
RP/0/0/CPU0:router(config)# extcommunity-set cost extcomm-cost  
RP/0/0/CPU0:router(config-ext)# IGP:90:914,  
RP/0/0/CPU0:router(config-ext)# Pre-Bestpath:91:915  
RP/0/0/CPU0:router(config-ext)# end-set
```

## extcommunity-set rt

To define a Border Gateway Protocol (BGP) route target (RT) extended community set, use the **extcommunity-set rt** command in global configuration mode. To remove the RT community set, use the **no** form of this command.

**extcommunity-set rt** *name*

**no extcommunity-set rt** *name*

### Syntax Description

<i>name</i>	Name of an RT extended community set.
-------------	---------------------------------------

### Command Default

No default behavior or values

### Command Modes

Global configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity-set</b> command.
Release 3.9.0	Support was added for more rt extended community formats.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity-set rt** command to define an RT extended community set for BGP.

Regular expressions and ranges can be specified to match the extended communities. Regular expressions and ranges can be specified in an extended community set to support the matching of communities. An attempt to use an extended community set that contains a range or regular expression to set an extended community set value is rejected when an attempt to attach such a policy is made.

An extcommunity set RT holds RT extended community values to match against the Border Gateway Protocol (BGP) RT extended community attribute. RT extended communities can be entered in these formats:

- *#-remark* ---Remark beginning with '#'
- *\*---* Wildcard (any community or part thereof)
- *1-4294967295---* 32-bit decimal number
- *1-65535 ---* 16-bit decimal number

- *A.B.C.D/M:N* ---Extended community - IPv4 prefix format
- *A.B.C.D:N*---Extended community - IPv4 format
- *ASN:N* ---Extended community - ASPLAIN format
- *X.Y:N* ---Extended community - ASDOT format
- **dfa-regex** ---DFA (deterministic finite automata) style regular expression
- **ios-regex** ---Traditional IOS style regular expression

**Note**

The dfa-regex and ios-regex syntax for community set is "[^<>]\*:[^<>]\*[ ]". This means that regex starts with a single-quote (") followed by a string of any character (that does not include single-quote, colon, ampersand, less-than, greater-than, or space) followed by a colon, and a string of any characters (that does not include single-quote, colon, ampersand, less-than, greater-than, or space) followed by single-quote.

*N* is a number within the range of 1 to 65535.

**Examples**

In the following example, an RT extended community set named extcomm-rt is defined:

```
RP/0/0/CPU0:router(config)# extcommunity-set rt extcomm-rt
RP/0/0/CPU0:router(config-ext)# 10002:666
RP/0/0/CPU0:router(config-ext)# 10.0.0.2:666
RP/0/0/CPU0:router(config-ext)# end-set
```

## extcommunity-set soo

To define a Border Gateway Protocol (BGP) Site-of-Origin (SoO) extended community set, use the **extcommunity-set soo** command in global configuration mode. To remove the SoO extended community set, use the **no** form of this command.

**extcommunity-set soo** *name*

**no extcommunity-set soo** *name*

### Syntax Description

<i>name</i>	Name of an SoO extended community set.
-------------	--

### Command Default

No default behavior or values

### Command Modes

Global configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity-set</b> command.
Release 3.9.0	Support was added for more soo extended community formats.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity-set soo** command to define an SoO extended community set.

An extcommunity set soo holds SoO extended community values to match against the Border Gateway Protocol (BGP) SoO extended community attribute. SoO extended communities can be entered in these formats:

- *#-remark* ---Remark beginning with '#'
- *\*---* Wildcard (any community or part thereof)
- *1-4294967295* ---32-bit decimal number
- *1-65535* ---16-bit decimal number
- *A.B.C.D/M:N* ---Extended community - IPv4 prefix format
- *A.B.C.D:N* ---Extended community - IPv4 format
- *ASN:N* ---Extended community - ASPLAIN format

- *X.Y:N* ---Extended community - ASDOT format
- **abort** ---Discard RPL definition and return to top level config
- **dfa-regex** ---DFA style regular expression
- **end-set** ---End of set definition
- **exit** ---Exit from the submode
- **ios-regex** ---Traditional IOS style regular expression
- **show** ---Show partial RPL configuration

*N* is a site-specific number.

## Examples

In the following example, a SoO extended community set named extcomm-soo is defined:

```
RP/0/0/CPU0:router(config)# extcommunity-set soo extcomm-soo
RP/0/0/CPU0:router(config-ext)# 66:60001,
RP/0/0/CPU0:router(config-ext)# 77:70001,
RP/0/0/CPU0:router(config-ext)# 88:80001,
RP/0/0/CPU0:router(config-ext)# 99:90001,

RP/0/0/CPU0:router(config-ext)# 100.100.100.1:153
RP/0/0/CPU0:router(config-ext)# end-set
```



## extcommunity soo is-empty

To determine if a Border Gateway Protocol (BGP) route has any Site-of-Origin (SoO) extended communities associated with it, use the **extcommunity soo is-empty** command in route-policy configuration mode.

**extcommunity soo is-empty**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Command History	Release	Modification
	Release 3.3.0	This command was introduced.
		This command replaced the <b>extcommunity is-empty</b> command.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity soo is-empty** command as a conditional expression within an **if** statement to check if a BGP SoO route has extended community attributes associated with it.



**Note** For a list of all conditional expressions available within an **if** statement, see the **if** command.

The **is-empty** operator takes no arguments and evaluates to true if the route has no SoO extended community attributes associated with it.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** In the following example, if a route has no SoO extended communities associated with it, the local preference is set to 100:

```
RP/0/0/CPU0:router(config)# route-policy extcommunity-is-empty-example
```

```
RP/0/0/CPU0:router(config-rpl)# if extcommunity soo is-empty then  
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## extcommunity soo matches-any

To match any element of a Border Gateway Protocol (BGP) Site-of-Origin (SoO) extended community set, use the **extcommunity soo matches-any** command in route-policy configuration mode.

**extcommunity soo matches-any** {*extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}

### Syntax Description

<i>extcommunity-set-name</i>	Name of a SoO extended community set.
<i>inline-extcommunity-set</i>	Inline SoO extended community set. The inline extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity matches-any</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity soo matches-any** command as a conditional expression within an **if** statement to match elements of an extended community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-any** operator evaluates as true if at least one extended community in the route matches an extended community specification in the named or inline set. If no extended community in the route matches any of the specifications in the named or inline set, then this simple condition evaluates to false. Likewise, when there is no extended community at all in the route, the condition evaluates to false.

Matching an extended community in the route to a specification in a named or an inline set is intuitive. In inline sets, extended community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, an SoO extended community set named extcomm-soo and a parameterized route policy named my-extcommunity-set-example(\$tag,\$ip) are defined.

The condition route policy named extcommunity soo matches-any is used in an if statement in this policy. If it evaluates to true, the local preference value is set to 100.

If it evaluates to false, the SoO extended community is evaluated using an inline set. If it evaluates to true, the local preference value is set to 200.

If it evaluates to false, the SoO extended community is evaluated using a different inline set. If it evaluates to true, the local preference value is set to 300.

If it evaluates to false, the SoO extended community is evaluated using a different inline set. If it evaluates to true, the local preference value is set to 400.

```
RP/0/0/CPU0:router(config)# extcommunity-set soo extcomm-soo
RP/0/0/CPU0:router(config-ext)# 66:60001,
RP/0/0/CPU0:router(config-ext)# 77:70001,
RP/0/0/CPU0:router(config-ext)# 88:80001,
RP/0/0/CPU0:router(config-ext)# 99:90001,
RP/0/0/CPU0:router(config-ext)# 100.100.100.1:153
RP/0/0/CPU0:router(config-ext)# end-set

RP/0/0/CPU0:router(config)# route-policy my-extcommunity-set-example($tag,$ip)
RP/0/0/CPU0:router(config-rpl)# if extcommunity soo matches-any extcomm-soo then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif extcommunity soo matches-any (10:20, 10:$tag)
then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# elseif extcommunity soo matches-any ($ip:$tag) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/0/CPU0:router(config-rpl-elseif)# elseif extcommunity soo matches-any (2.3.4.5:$tag)
then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 400
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">extcommunity rt matches-any, on page 61</a>	Matches any element of a BGP RT extended community set.
<a href="#">extcommunity soo matches-every, on page 77</a>	Matches every element of a BGP SoO extended community set.

## extcommunity soo matches-every

To match every element of a Border Gateway Protocol (BGP) Site-of-Origin (SoO) extended community set, use the **extcommunity soo matches-every** command in route-policy configuration mode.

**extcommunity soo matches-every** {*extcommunity-set-name*| *inline-extcommunity-set*| *parameter*}

### Syntax Description

<i>extcommunity-set-name</i>	Name of a SoO extended community set.
<i>inline-extcommunity-set</i>	Inline SoO extended community set. The inline extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>extcommunity matches-every</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **extcommunity soo matches-every** command as a conditional expression within an **if** statement to match every element of a SoO extended community set.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A simple condition using the **matches-every** operator evaluates as true if every extended community value in the extended community attribute for the route matches at least one element of the extended community set or inline set. If no extended community in the route matches any of the specifications in the named or inline set, then this simple condition evaluates to false. Likewise, when there is no extended community at all in the route, the condition evaluates to false.

Matching an extended community in the route to a specification in a named or an inline set is intuitive. In inline sets, extended community specifications may be parameterized, in which case the relevant matching is done when the value of the parameter has been supplied.

## Examples

In the following example, an extended community set named my-extcomm-rt-set and a parameterized route policy named extcommunity-matches-every-example(\$as, \$tag) are defined. The condition extcommunity soo matches-every is used in an if statement in this policy and if it evaluates to true, the local-preference value is set to 100. If it evaluates to false, the extended community is evaluated using an inline set. If that condition evaluates to true, the local-preference value is set to 200. If it evaluates to false, the local-preference value is set to 300.

```
RP/0/0/CPU0:router(config)# extcommunity-set soo my-extcomm-rt-set
RP/0/0/CPU0:router(config-ext)# 10:20,
RP/0/0/CPU0:router(config-ext)# 10:30,
RP/0/0/CPU0:router(config-ext)# 10:40
RP/0/0/CPU0:router(config-ext)# end-set

RP/0/0/CPU0:router(config)# route-policy extcommunity-matches-every-example($as, $tag)
RP/0/0/CPU0:router(config-rpl)# if extcommunity soo matches-every my-extcomm-rt-set then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)# elseif extcommunity soo matches-every (10:20, 10:$tag,
$as:30) then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)# else
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/0/CPU0:router(config-rpl-elseif)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">extcommunity soo matches-any, on page 75</a>	Matches any element of a BGP SoO extended community set.

# globalVarN is

To check the value of globalVarN value assigned through the **var globalVarN** command, use the **globalVarN is** command in router-policy configuration mode.

**globalVarN is** {*number* | *parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is from 1 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

If the globalVarN is not assigned using the **var globalVarN** command, then the default value for globalVarN is zero.

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 5.1.3	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

The scope of this variable is from the start of policy to end of policy attached under any attach-point. The variable can be assigned in any child policy and can check for value in parent policy or any other hierarchy of route-policy referred using apply statement.

## Task ID

Task ID	Operation
route-policy	read, write

## Examples

This example is to identify routes which has communities configured under community-sets internal\_set and service2 and set different local-preference values based on the community values.

```
community-set internal_set
  1:1
end-set
community-set service2
  3:3
```

```
end-set
route-policy child1
  var globalVar1 100
  #... user can add more actions or conditions...
end-policy
route-policy child2
  var globalVar2 200
  #...user can add more actions or conditions ..
end-policy
route-policy inbound
  if community matches-any internal_set then
    apply child1
  endif
  if community matches-any service2 then
    apply child2
  endif
  if globalVar1 is 100 and globalVar2 is 200 then
    set local-preference 250
  elseif globalVar1 is 100 then
    set local-preference 150
  elseif globalVar2 is 200 then
    set local-preference 50
  endif
end-policy
```



# if

To decide which actions or dispositions should be taken for a given route, use the **if** command in route-policy configuration mode.

**if** *conditional-expression* **then** *action-statement* [ *action-statement* ] [**elseif** *conditional-expression* **then** *action-statement* [ *action-statement* ] ] [**else** *action-statement* [ *action-statement* ] ] **endif**

## Syntax Description

<i>conditional-expression</i>	Expression to decide which actions or dispositions should be taken for the given route.
<b>then</b>	Executes an action statement if the <b>if</b> condition is true.
<b>elseif</b>	Strings together a sequence of tests.
<b>else</b>	Executes an action statement if the <b>if</b> condition is false.
<b>endif</b>	Ends the <b>if</b> statement.
<i>action-statement</i>	Sequence of operations that modify a route.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 4.2.0	Support was added for Apply Condition Policies that allow the usage of a route-policy in an "if" statement of another route-policy.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

The **if** command uses a conditional expression to decide which actions or dispositions should be taken for a given route. [Table 1: Conditional Expressions, on page 82](#) lists the conditional expressions.

An action statement is a sequence of operations that modify a route, most of which are distinguished by the **set** keyword. In a route policy, these operations can be grouped. [Table 2: Action Statements, on page 84](#) lists the action statements.

Apply Condition policies allow usage of a route-policy in an "if" statement of another route-policy.

```
Route-policy policy_name
If apply policyA and apply policyB then
Set med 100
Else if not apply policyD then
Set med 200
Else
Set med 300
Endif
End-policy
```

**Table 1: Conditional Expressions**

Command	Description
<a href="#">as-path in, on page 12</a>	Matches the AS path of a route to an AS path set. The AS path is a sequence of autonomous system numbers traversed by a route.
<a href="#">as-path is-local, on page 14</a>	Determines if the router (or another router within this autonomous system or confederation) originated the route.
<a href="#">as-path length, on page 16</a>	Performs a conditional check based on the length of the AS path.
<a href="#">as-path neighbor-is, on page 18</a>	Tests the autonomous system number or numbers at the head of the AS path against a sequence of one or more integral values or parameters.
<a href="#">as-path originates-from, on page 21</a>	Tests an AS path against the AS sequence beginning with the AS number that originated a route.
<a href="#">as-path passes-through, on page 24</a>	Tests to learn if the specified integer or parameter appears anywhere in the AS path or if the sequence of integers and parameters appears.
<a href="#">as-path unique-length, on page 29</a>	Performs specific checks based on the length of the AS path.
<a href="#">community is-empty, on page 31</a>	Learns if a route has community attributes associated with it.
<a href="#">community matches-any, on page 32</a>	Matches any element of a community set.
<a href="#">community matches-every, on page 35</a>	Matches every element of a community set.
<a href="#">destination in, on page 45</a>	Matches a destination entry in a named prefix set or inline prefix set.
<a href="#">extcommunity rt is-empty, on page 59</a>	Learns if a route has RT extended community attributes associated with it.
<a href="#">extcommunity rt matches-any, on page 61</a>	Matches elements of an RT extended community set.

Command	Description
<a href="#">extcommunity rt matches-every, on page 63</a>	Matches every element of an RT extended community set.
<a href="#">extcommunity rt matches-within, on page 65</a>	Matches at least one element of a Border Gateway Protocol (BGP) route target (RT) extended community set.
<a href="#">extcommunity soo is-empty, on page 73</a>	Learns if a route has SoO extended community attributes associated with it.
<a href="#">extcommunity soo matches-any, on page 75</a>	Matches elements of an SoO extended community set.
<a href="#">extcommunity soo matches-every, on page 77</a>	Matches every element of an SoO extended community set.
<a href="#">local-preference, on page 88</a>	Specifies BGP local-preference attribute
<a href="#">med, on page 90</a>	Compares the MED to an integer value or a parameterized value.
<a href="#">next-hop in, on page 92</a>	Compares the next-hop associated with the route to data contained in either a named or an inline prefix set.
<a href="#">orf prefix in, on page 94</a>	Matches a prefix in a prefix set or an inline prefix set.
<a href="#">origin is, on page 96</a>	Tests the value of the origin attribute.
<a href="#">path-type is, on page 100</a>	Tests the path type.
<a href="#">protocol, on page 109</a>	Checks if a protocol is installing the route.
<a href="#">rd in, on page 111</a>	Compares the RD associated with the route to data contained in either a named or an inline RD set.
<a href="#">rib-has-route, on page 117</a>	Checks if a route is in the RIB.
<a href="#">route-has-label, on page 119</a>	Checks if a route has a Multiprotocol Label Switching (MPLS) label.
<a href="#">route-type is, on page 122</a>	Compares route types when redistribution is being performed into BGP, OSPF, or IS-IS.
<a href="#">source in, on page 286</a>	Tests the source of the route against the data in either a named or an inline prefix set.
<a href="#">tag, on page 290</a>	Matches a specific tag value.

Command	Description
<a href="#">tag in, on page 292</a>	Conditionally compares tag-route against tag-set.
<a href="#">vpn-distinguisher is, on page 300</a>	Compares the VPN distinguisher against a specified value.

**Table 2: Action Statements**

Command	Description
<a href="#">abort (RPL), on page 6</a>	Discards a route policy definition and returns to global configuration mode.
<a href="#">add, on page 8</a>	Adds an offset to an existing value.
<a href="#">apply, on page 10</a>	Executes a parameterized or an unparameterized policy from within another policy.
<a href="#">delete community, on page 41</a>	Deletes community values from a community list in a route.
<a href="#">delete extcommunity rt, on page 43</a>	Deletes extended community values from an extended community list in a route.
<a href="#">done, on page 47</a>	Accepts this route with no further processing
<a href="#">drop, on page 49</a>	Drops a route.
<a href="#">end-policy, on page 55</a>	Ends the definition of a route policy and exits route-policy configuration mode.
<a href="#">pass, on page 98</a>	Signifies that even though the route has not been modified, the user wants to continue executing in the policy block.
<a href="#">prepend as-path, on page 107</a>	Prepends the AS path with additional autonomous system numbers.
<a href="#">replace as-path, on page 115</a>	Replaces a sequence of AS numbers or private AS numbers in the AS path with the configured local AS.
<a href="#">set community, on page 128</a>	Sets the BGP community attribute.
<a href="#">set dampening, on page 132</a>	Configures BGP route dampening.
<a href="#">set eigrp-metric, on page 134</a>	Sets the Enhanced Interior Gateway Routing Protocol (EIGRP) metric value.

Command	Description
<a href="#">set extcommunity cost, on page 136</a>	Replaces or adds the extended communities for a cost on the route.
<a href="#">set extcommunity rt, on page 138</a>	Replaces or adds the extended communities for an RT on the route.
<a href="#">set ip-precedence, on page 140</a>	Sets the IP precedence to classify packets.
<a href="#">set isis-metric, on page 142</a>	Sets the IS-IS metric attribute value.
<a href="#">set label, on page 143</a>	Sets the BGP label attribute value.
<a href="#">set level, on page 146</a>	Configures the IS-IS level in which redistributed routes should be sent.
<a href="#">set local-preference, on page 148</a>	Specifies a preference value for the autonomous system path.
<a href="#">set med, on page 150</a>	Sets the MED value.
<a href="#">set metric-type (IS-IS), on page 152</a>	Controls whether IS-IS treats the metric as an internal or external metric.
<a href="#">set metric-type (OSPF), on page 154</a>	Controls whether OSPF treats the cost as a Type 1 or Type 2 metric.
<a href="#">set next-hop, on page 156</a>	Replaces the next-hop associated with a given route.
<a href="#">set origin, on page 158</a>	Changes the origin attribute.
<a href="#">set ospf-metric, on page 160</a>	Sets an OSPF protocol metric attribute value.
<a href="#">set qos-group (RPL), on page 161</a>	Sets the QoS group to classify packets.
<a href="#">set rib-metric, on page 163</a>	Sets a RIB metric attribute value for a table policy.
<a href="#">set rip-metric, on page 164</a>	Sets RIP metric attributes.
<a href="#">set rip-tag, on page 166</a>	Sets route tag attribute.
<a href="#">set tag, on page 173</a>	Sets the tag attribute.
<a href="#">set traffic-index, on page 175</a>	Sets the traffic index attribute.
<a href="#">set weight, on page 179</a>	Sets the weight value for BGP routes.
<a href="#">suppress-route, on page 288</a>	Indicates that a given component of an aggregate should be suppressed, that is, not advertised.

Command	Description
<a href="#">unsuppress-route</a> , on page 296	Indicates that a given component of an aggregate should be unsuppressed.
<a href="#">set vpn-distinguisher</a> , on page 177	Sets the VPN distinguisher value.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, any route whose AS path is in the set as-path-set-1 is dropped:

```
RP/0/0/CPU0:router(config-rpl)# if as-path in as-path-set-1 then
RP/0/0/CPU0:router(config-rpl-if)# drop
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

The contents of the **then** clause may be an arbitrary sequence of action statements.

The following example shows an **if** statement with two action statements:

```
RP/0/0/CPU0:router(config-rpl)# if origin is igp then
RP/0/0/CPU0:router(config-rpl-if)# set med 42
RP/0/0/CPU0:router(config-rpl-if)# prepend as-path 73 5
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

The **if** command also permits an **else** clause to be executed if the expression is false, as follows:

```
RP/0/0/CPU0:router(config-rpl)# if med eq 200 then
RP/0/0/CPU0:router(config-rpl-if)# set community (12:34) additive
RP/0/0/CPU0:router(config-rpl-if)# else
RP/0/0/CPU0:router(config-rpl-else)# set community (12:56) additive
RP/0/0/CPU0:router(config-rpl-else)# endif
RP/0/0/CPU0:router(config-rpl)#
```

The routing policy language (RPL) also provides syntax using the **elseif** command to string together a sequence of tests, as shown in the following example:

```
RP/0/0/CPU0:router(config-rpl)# if med eq 150 then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 10
RP/0/0/CPU0:router(config-rpl-if)# elseif med eq 200 then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 60
RP/0/0/CPU0:router(config-rpl-elseif)# elseif med eq 250 then
RP/0/0/CPU0:router(config-rpl-elseif)# set local-preference 110
RP/0/0/CPU0:router(config-rpl-elseif)# else
RP/0/0/CPU0:router(config-rpl-else)# set local-preference 0
RP/0/0/CPU0:router(config-rpl-else)# endif
RP/0/0/CPU0:router(config-rpl)#
```

The statements within an **if** statement may themselves be **if** statements, as shown in this example:

```
RP/0/0/CPU0:router(config-rpl)# if community matches-any (12:34, 56:78) then
RP/0/0/CPU0:router(config-rpl-if)# if med eq 150 then
```

```
RP/0/0/CPU0:router(config-rpl-if)# drop  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

The policy configuration shown sets the value of the local preference attribute to 100 on any route that has a community value of 12:34 or 56:78 associated with it. However, if any of these routes has a Multi Exit Discriminator (MED) value of 150, then each route with both the community value of 12:34 or 56:78 and a MED of 150 is dropped.

# local-preference

To compare the local-preference attribute of a BGP route to an integer value or a parameterized value, use the local-preference command in route-policy configuration mode.

**local-preference** {**eq**|**is**|**ge**|**le**} {*number*|*parameter*}

## Syntax Description

<b>eq</b>   <b>is</b>   <b>ge</b>   <b>le</b>	Equal to; exact match; greater than or equal to; less than or equal to.
<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **local-preference** command as a conditional expression within an **if** statement to compare the local-preference attribute to an integer value or a parameterized value.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The MED is a 32-bit unsigned integer. The **eq** operation compares the local-preference to either a static value or a parameterized value passed to a parameterized policy for equality with that value. A greater than or equal to comparison can also be done with the **ge** operator, and a less than or equal to comparison can be performed using the **le** operator.

## Examples

The following example shows that if the **local-preference** is 10, local-preference is set to 100:

```
RP/0/RSP0RP0/CPU0:router(config-rpl)# if local-preference eq 10 then
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# set weight 100
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# endif
```



```
RP/0/RSP0RP0/CPU0:router(config-rpl)#
```

# med

To compare the Multi Exit Discriminator (MED) to an integer value or a parameterized value or compare the MED attribute of a BGP route to an integer value, use the **med** command in route-policy configuration mode.

**med** {**eq**|**is**|**ge**|**le**} {*number*|*parameter*}

## Syntax Description

<b>eq</b>   <b>is</b>   <b>ge</b>   <b>le</b>	Equal to; exact match; greater than or equal to; less than or equal to.
<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **med** command as a conditional expression within an **if** statement to compare the MED to an integer value or a parameterized value.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The MED is a 32-bit unsigned integer. The **eq** operation compares the MED to either a static value or a parameterized value passed to a parameterized policy for equality with that value. A greater than or equal to comparison can also be done with the **ge** operator, and a less than or equal to comparison can be performed using the **le** operator.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows that if the **med** commands match, the local preference is set to 100:

```
RP/0/0/CPU0:router(config-rpl)# if med eq 10 then  
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 100  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

## next-hop in

To compare the next-hop associated with the route to data contained in either an inline or a named prefix set, use the **next-hop in** command in route-policy configuration mode.

**next-hop in** {*prefix-set-name*| *inline-prefix-set*| *parameter*}

### Syntax Description

<i>prefix-set-name</i>	Name of a prefix set.
<i>inline-prefix-set</i>	Inline prefix set. The inline prefix set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **next-hop in** command as a conditional expression within an **if** statement to compare the next-hop associated with the route to data contained in either an inline or a named prefix set. The result is true if any value in the prefix set matches the next-hop of the route. A comparison that refers to a named prefix set that has no elements in it returns false.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The next-hop is an IPv4 address entered as a dotted-decimal or an IPv6 address entered as a colon-separated hexadecimal.

### Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows that if the **next-hop in** commands match, the local preference is set to 100

```
RP/0/0/CPU0:router(config-rpl)# if next-hop in some-prefix-set then
RP/0/0/CPU0:router(config-rpl-if)# if next-hop in (10.0.0.5, fe80::230/64) then
RP/0/0/CPU0:router(config-rpl-if)# set local-preference 0
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

# orf prefix in

To configure an outbound route filter (ORF), use the **orf prefix in** command in route-policy configuration mode.

**orf prefix in** {*prefix-set-name*| *inline-prefix-set*}

## Syntax Description

<i>prefix-set-name</i>	Name of a prefix set.
<i>inline-prefix-set</i>	Inline prefix set. The inline prefix set must be enclosed in parentheses.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **orf prefix in** command to match a prefix in a prefix set or an inline prefix set.

This command takes either a named prefix set or an inline prefix set value as an argument. It returns true if the destination NLRI matches any entry in the prefix set. An attempt to match destination using a prefix set that is defined but contains no elements returns false.

This command is used in the context of the orf route-policy attach point in BGP. The destination of a route is also known in Border Gateway Protocol (BGP) as its network-layer reachability information (NLRI). It comprises a prefix value and a mask length. The routing policy language (RPL) provides one operation on prefixes, testing them for matching against a list of prefix-match specifications using the **in** operator.

## Examples

In the following example, the prefix set orfpreset1 and the route policy named orfpolicy are defined. Next, the orfpolicy is applied to the neighbor orf attach point.

If the prefix of the route matches any of the prefixes specified in orfpreset1 (211.105.1.0/24, 211.105.5.0/24, 211.105.11.0/24), then the prefix is dropped. If the prefix matches in(211.105.3.0/24, 211.105.7.0/24, 211.105.13.0/24), then the prefix is accepted. In addition to this inbound filtering, BGP sends these prefix

entries to the upstream neighbor indicating a permit or deny so that the neighbor can make the same filter updates.

```
RP/0/0/CPU0:router(config)# prefix-set orfpreset1
RP/0/0/CPU0:router(config-pfx)# 211.105.1.0/24,
RP/0/0/CPU0:router(config-pfx)# 211.105.5.0/24,
RP/0/0/CPU0:router(config-pfx)# 211.105.11.0/24
RP/0/0/CPU0:router(config-pfx)# end-set
!
!
RP/0/0/CPU0:router(config)# route-policy orfpolicy
RP/0/0/CPU0:router(config-rpl)# if orf prefix in orfpreset1 then
RP/0/0/CPU0:router(config-rpl-if)# drop
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# if orf prefix in (211.105.3.0/24, 211.105.7.0/24,
211.105.13.0/24) then
RP/0/0/CPU0:router(config-rpl-if)# pass
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
!
!
RP/0/0/CPU0:router(config)# router bgp 2
RP/0/0/CPU0:router(config-bgp)# neighbor 1.1.1.1
RP/0/0/CPU0:router(config-bgp-nbr)# remote-as 3
RP/0/0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/0/CPU0:router(config-bgp-nbr-af)# orf route-policy orfpolicy
```

#### Related Commands

Command	Description
<b>orf</b>	Specifies BGP ORF and inbound filtering criteria.

# origin is

To match a specific origin type, use the **origin is** command in route-policy configuration mode.

**origin is** {**igp**|**egp**|**incomplete**|*parameter*}

## Syntax Description

<b>igp</b>	Specifies Interior Gateway Protocol.
<b>egp</b>	Specifies Exterior Gateway Protocol.
<b>incomplete</b>	Specifies that Border Gateway Protocol (BGP) first learned the route by means other than BGP or Interior Gateway Protocol (IGP); for example, the route is learned through configuration.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **origin is** command as a conditional expression within an **if** statement to test the value of the origin attribute.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The origin of a BGP route is an enumeration; it is **igp** , **egp** , or **incomplete** .

This command can be parameterized.

## Task ID

Task ID	Operations
route-policy	read, write



## Examples

In the following example, the origin is tested within an **if** statement to learn if it is either **igp** or **egp** :

```
RP/0/0/CPU0:router(config-rpl)# if origin is igp or origin is egp then
```

In the following example, a parameter is used to match a specific origin type:

```
RP/0/0/CPU0:router(config)# route-policy bar($origin)
RP/0/0/CPU0:router(config-rpl)# if origin is $origin then
RP/0/0/CPU0:router(config-rpl-if)# set med 20
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

# pass

To pass a route for further processing, use the **pass** command in route-policy configuration mode.

**pass**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **pass** command to signify that even though this route has not been modified, the user wants to continue executing in this policy block.



### Note

The **pass** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

When a policy block has finished executing, any route that has been modified in this policy block or has received a pass disposition in this policy block passes the policy and execution finishes for that policy. If this policy block is applied from within another policy block and the route is either passed or modified, then execution continues in the policy block that applied this policy block.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows how to accept the route unconditionally without modifying it:

```
RP/0/0/CPU0:router(config-rpl)# pass
```

This example accepts the route unconditionally, without modifying it, if the destination is in prefix-set permitted:

```
RP/0/0/CPU0:router(config-rpl)# if destination in permitted then  
RP/0/0/CPU0:router(config-rpl-if)# pass  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

# path-type is

To match path types, use the **path-type is** command in route-policy configuration mode.

**path-type is** {**ibgp**|**ebgp**|*parameter*}

## Syntax Description

<b>ibgp</b>	Specifies an internal BGP path.
<b>ebgp</b>	Specifies an external BGP path.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **path-type is** command as a conditional expression within an **if** statement to match path types.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, if the path is an external BGP path the route is accepted:

```
RP/0/0/CPU0:router(config)# route-policy policy_A
RP/0/0/CPU0:router(config-rpl)# if path-type is ebgp then
RP/0/0/CPU0:router(config-rpl-if)# pass
```

```
RP/0/0/CPU0:router(config-rpl-if)# else  
RP/0/0/CPU0:router(config-rpl-else)# drop  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# policy-global

To define global parameters and enter global parameter configuration mode, use the **policy-global** command in global configuration mode. To remove global parameters, use the **no** form of this command.

**policy-global**

**no policy-global**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Global configuration

Command History	Release	Modification
	Release 3.4.0	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **policy-global** command to define global parameters and enter global parameter configuration mode.

RPL supports the definition of systemwide global parameters that can be used inside a policy definition. The global parameter values can be used directly inside a policy definition similar to the local parameters of parameterized policy. When a parameterized policy has a parameter name “collision” with a global parameter name, parameters local to policy definition take precedence, effectively ‘masking off’ global parameters. In addition, a validation mechanism is in place to prevent the deletion of a particular global parameter if it is referred by any policy. For more information on global parameters and parameterization, see the *Implementing Routing Policy* module of the *Cisco IOS XR Routing Configuration Guide for the Cisco XR 12000 Series Router*.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** The following example shows how to configure global parameters:

```
RP/0/0/CPU0:router(config)# policy-global
RP/0/0/CPU0:router(config-rp-gl)# glbpathtype 'ebgp'
```

```
RP/0/0/CPU0:router(config-rp-gl)# glbtag '100'  
RP/0/0/CPU0:router(config-rp-gl)# end-global
```

In the following example, the *globalparam* argument makes use of the global parameters *gbpathtype* and *glbtag* defined above and is defined for a nonparameterized policy:

```
RP/0/0/CPU0:router(config)# route-policy globalparam  
RP/0/0/CPU0:router(config-rpl)# if path-type is $glbpathtype then  
RP/0/0/CPU0:router(config-rpl)# set tag $glbtag  
RP/0/0/CPU0:router(config-rpl)# endif  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

#### Related Commands

Command	Description
<a href="#">end-global</a> , on page 54	Ends the definition of global parameters.

# prefix-set

To enter prefix set configuration mode and define a prefix set for contiguous and non-contiguous set of bits, use the **prefix-set** command in global configuration mode. To remove a named prefix set, use the **no** form of this command.

**prefix-set** *name*

**no prefix-set** *name*

## Syntax Description

<i>name</i>	Name of a prefix set.
-------------	-----------------------

## Command Default

None

## Command Modes

Global configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 5.1.1	The command was modified to accept IPv4 and IPv6 address and address mask to define a prefix set.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **prefix-set** command to enter prefix set configuration mode and define a prefix set.

A prefix set is a comma-separated list of prefix match specifications. It holds IPv4 or IPv6 prefix match specifications, each of which has four parts: an address, a mask length, a minimum matching length, and a maximum matching length. The address is required, but the other three parts are optional. The address is a standard four-part, dotted-decimal numeric IPv4 address or a colon-separated hexadecimal IPv6 address. The mask length, if present, is a nonnegative decimal integer in the range from 0 to 32 for IPv4 prefixes or 0 to 128 for IPv6 prefixes following the address and separated from it by a slash. The optional minimum matching length follows the address and optional mask length and is expressed as the keyword **ge** (mnemonic for **greater than or equal to**), followed by a nonnegative decimal integer in the range from 0 to 32 for IPv4 or 0 to 128 for IPv6. The optional maximum matching length follows the rest and is expressed by the keyword **le** (mnemonic for **less than or equal to**), followed by yet another nonnegative decimal integer in the range from 0 to 32 for IPv4 or 0 to 128 for IPv6. A syntactic shortcut for specifying an exact length for prefixes to match is the **eq** keyword, mnemonic for **equal to**.

If a prefix match specification has no mask length, then the default mask length is 32 for IPv4 or 128 for IPv6. The default minimum matching length is the mask length. If a minimum matching length is specified, then



the default maximum matching length must be less than 32 for IPv4 prefixes or 128 for IPv6 prefixes. Otherwise, if neither a minimum nor maximum length is specified, the default maximum length is the mask length.

A prefix set is a list of prefix match specifications. It holds IPv4 or IPv6 prefix match specifications, each of which has two parts: an address and a mask. The address and mask is a standard dotted-decimal IPv4 or colon-separated hexadecimal IPv6 address. The prefix set allows the specifying of contiguous and non-contiguous set of bits that must be matched in any route. The set of bits to be matched are provided in the form of a mask in which a binary 0 means a mandatory match and a binary 1 means a 'do not match' condition.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows a prefix set named legal-ipv4-prefix-examples:

```
RP/0/0/CPU0:router(config)# prefix-set legal-ipv4-prefix-examples
RP/0/0/CPU0:router(config-pfx)# 10.0.1.1,
RP/0/0/CPU0:router(config-pfx)# 10.0.2.0/24,
RP/0/0/CPU0:router(config-pfx)# 10.0.3.0/24 ge 28,
RP/0/0/CPU0:router(config-pfx)# 10.0.4.0/24 le 28,
RP/0/0/CPU0:router(config-pfx)# 10.0.5.0/24 ge 26 le 30,
RP/0/0/CPU0:router(config-pfx)# 10.0.6.0/24 eq 28
RP/0/0/CPU0:router(config-pfx)# end-set
```

The first element of the prefix set matches only one possible value, 10.0.1.1/32 or the host address 10.0.1.1. The second element matches only one possible value, 10.0.2.0/24. The third element matches a range of prefix values, from 10.0.3.0/28 to 10.0.3.255/32. The fourth element matches a range of values, from 10.0.4.0/24 to 10.0.4.240/28. The fifth element matches prefixes in the range from 10.0.5.0/26 to 10.0.5.252/30. The sixth element matches any prefix of length 28 in the range from 10.0.6.0/28 through 10.0.6.240/28.

The following prefix set consists entirely of invalid prefix match specifications:

```
RP/0/0/CPU0:router(config)# prefix-set INVALID-PREFIX-EXAMPLES
RP/0/0/CPU0:router(config-pfx)# 10.1.1.1 ge 16,
RP/0/0/CPU0:router(config-pfx)# 10.1.2.1 le 16,
RP/0/0/CPU0:router(config-pfx)# 10.1.3.0/24 le 23,
RP/0/0/CPU0:router(config-pfx)# 10.1.4.0/24 ge 33,
RP/0/0/CPU0:router(config-pfx)# 10.1.5.0/25 ge 29 le 28
RP/0/0/CPU0:router(config-pfx)# end-set
```

Neither the minimum length nor the maximum length is legal without a mask length. The maximum length must be at least the mask length. The minimum length must be less than 32, the maximum length of an IPv4 prefix. The maximum length must be equal to or greater than the minimum length.

The following example shows a valid IPv6 prefix set named legal-ipv6-prefix-examples:

```
RP/0/0/CPU0:router(config)# prefix-set legal-ipv6-prefix-examples
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:1::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:2::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:3::/64,
RP/0/0/CPU0:router(config-pfx)# 2001:0:0:4::/64
RP/0/0/CPU0:router(config-pfx)# end-set
```

## Examples

This example shows a prefix set named legal-ipv4-prefix:

```
RP/0/0/CPU0:router(config)# prefix-set legal-ipv4-prefix  
RP/0/0/CPU0:router(config-pfx)# 10.1.1.1 0.255.0.255  
RP/0/0/CPU0:router(config-pfx)# 10.2.2.2 0.0.0.0  
RP/0/0/CPU0:router(config-pfx)# 10.3.3.3 255.255.255.254  
RP/0/0/CPU0:router(config-pfx)# 10.4.4.4 255.255.255.255
```

In the above example, In the above example, the command defines the prefix-set named acl-prefix-set. The first element specifies to match all routes having 10 in first octet and 1 in third octet. The second element matches all routes having prefix as 10.2.2.2 (that is, matches all conditions). The third element matches all routes having odd numbers in the last octets and the fourth element matches all routes with any prefix.

# prepend as-path

To prepend the AS path with additional autonomous system numbers, use the **prepend as-path** command in route-policy configuration mode.

**prepend as-path** {*as-number*|*parameter*| **most-recent**} [*number*|*parameter*]

## Syntax Description

<i>as-number</i>	Autonomous system number to prepend to the path. <ul style="list-style-type: none"> <li>• Range for 2-byte Autonomous system numbers (ASNs) is 1 to 65535.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asplain format is 1 to 4294967295.</li> <li>• Range for 4-byte Autonomous system numbers (ASNs) in asdot format is 1.0 to 65535.65535.</li> </ul>
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>most-recent</b>	Specifies that the most recent autonomous system number should be prepended.
<i>number</i>	(Optional) Number of times the autonomous system number should be prepended. Range is 1 to 63.

## Command Default

The default *number* is 1.

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>as-number</i> 4-byte number range 1.0 to 65535.65535 was supported. The <b>most-recent</b> keyword was added.
Release 3.9.0	Asplain format for 4-byte Autonomous system numbers notation was supported.

## Usage Guidelines

Use the **prepend as-path** command to prepend the AS path with additional autonomous system numbers.

**Note**

The **prepend as-path** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

This command can take one or two arguments. The first argument (either a number or parameter) is the autonomous system number to prepend to the path. The optional second argument (either a number or parameter) is the number of times the autonomous system number should be prepended.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following example shows how to prepend the autonomous system number 666.1 to the AS path three times:

```
RP/0/0/CPU0:router(config-rpl)# prepend as-path 666.1 3
```

The following example shows how to prepend the autonomous system number 666.0 to the AS path one time:

```
RP/0/0/CPU0:router(config-rpl)# prepend as-path 666.0 1
```

# protocol

To check the protocol that installs the route, use the **protocol** command in route-policy configuration mode.

**protocol** {**in** | (*protocol-set*) | **is** | *protocol-name*}

## Syntax Description

<b>in</b> ( <i>protocol-set</i> )	<p>Specifies a member of a set. The <i>protocol-set</i> argument accepts the following keywords within parentheses:</p> <ul style="list-style-type: none"> <li>• <b>bgp</b> —Border Gateway Protocol (BGP)</li> <li>• <b>connected</b> —Connected routes</li> <li>• <b>eigrp</b> —Enhanced Interior Gateway Routing Protocol (EIGRP)</li> <li>• <b>isis</b> —ISO Intermediate System-to-Intermediate System (IS-IS)</li> <li>• <b>ospf</b> —Open Shortest Path First (OSPF)</li> <li>• <b>ospfv3</b> —Open Shortest Path First version 3 (OSPFv3)</li> <li>• <b>rip</b> —Routing Information Protocol (RIP)</li> <li>• <b>static</b> —Static routes</li> </ul> <p>Keywords must be separated by a comma.</p>
<b>is</b> <i>protocol-name</i>	Specifies a single protocol name, and accepted keywords are similar to the <i>protocol-set</i> argument.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
Release 3.6.0	Support for the <b>ospfv3</b> keyword was added to the <i>protocol-set</i> argument.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **protocol** command as a conditional expression within an if statement to specify a protocol to install a route.

Use the **in** keyword to determine if a protocol listed in the *protocol-set* is the originator of the route being filtered.

Use the **is** keyword to determine if *protocol-name* is an exact match.

**Note**

For a list of all conditional expressions available within an **if** statement, see the **if** command.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following example shows how to use the **protocol** command as a conditional expression within if statements:

```
RP/0/0/CPU0:router(config)# route-policy rip1
RP/0/0/CPU0:router(config-rpl)# if protocol in (connected, static) then
RP/0/0/CPU0:router(config-rpl-if)# add rip-metric 2
RP/0/0/CPU0:router(config-rpl-if)# elseif protocol is bgp 1 then
RP/0/0/CPU0:router(config-rpl-elseif)# add rip-metric 3
RP/0/0/CPU0:router(config-rpl-elseif)# elseif protocol is ospf 2 then
RP/0/0/CPU0:router(config-rpl-elseif)# add rip-metric 4
RP/0/0/CPU0:router(config-rpl-elseif)# else
RP/0/0/CPU0:router(config-rpl-else)# add rip-metric 5
RP/0/0/CPU0:router(config-rpl-else)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy

RP/0/0/CPU0:router(config)# router rip
RP/0/0/CPU0:router(config-rip)# interface GigabitEthernet0/1/0/1
RP/0/0/CPU0:router(config-rip-if)# route-policy rip1 out
```

# rd in

To compare the route distinguisher (RD) associated with the route to RDs contained in either a named or an inline RD set, use the **rd in** command in route-policy configuration mode.

**rd in** {*rd-set-name*|*inline-rd-set*|*parameter*}

## Syntax Description

<i>rd-set-name</i>	Name of an RD set.
<i>inline-rd-set</i>	Inline RD set. The inline RD set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **rd in** command as a conditional expression within an **if** statement to match a destination entry in a named prefix set or inline prefix set.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command takes either a named RD set or an inline RD set value as an argument. The condition returns true if the destination entry matches any entry in the RD set or inline RD set. An attempt to match an RD using an RD set that is defined but contains no elements returns false.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows the **rd in** command with an inline RD set value as an argument:

```
RP/0/0/CPU0:router(config)# route-policy
RP/0/0/CPU0:router(config-rpl)# if rd in (128.1.0.0/16:100) then
RP/0/0/CPU0:router(config-rpl-if)# pass
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```



# rd-set

To define a route distinguisher (RD) set and enter RD configuration mode, use the **rd-set** command in global configuration mode.

**rd-set** *name*

**no rd-set** *name*

## Syntax Description

name	Name of an RD community set.
------	------------------------------

## Command Default

No default behavior or values

## Command Modes

Global configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **rd-set** command to create a set with RD elements and enter RD configuration mode. An RD set is a 64-bit value prepended to an IPv4 address to create a globally unique Border Gateway Protocol (BGP) VPN IPv4 address.



### Note

For *m*, the mask length is supported.

You can define RD values with the following commands:

- *a.b.c.d/m:\**—BGP VPN RD in IPv4 format with a wildcard character. For example, 10.0.0.2/24.0:\*
- *a.b.c.d/m:n*—BGP VPN RD in IPv4 format with a mask. For example, 10.0.0.2/24:666.
- *a.b.c.d:\** —BGP VPN RD in IPv4 format with a wildcard character. For example, 10.0.0.2:\*
- *a.b.c.d:n*—BGP VPN RD in IPv4 format. For example, 10.0.0.2:666.
- *asn:\**—BGP VPN RD in ASN format with a wildcard character. For example, 10002:\*
- *asn:n*—BGP VPN RD in ASN format. For example, 10002:666.
- *x.y:\**—BGP VPN RD in 4-byte ASN format with a wildcard character. For example, 10002.101:\*

- *x.y:n*—BGP VPN RD in 4-byte ASN format. For example, 10002.101:666.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following example shows how to create an RD set called my\_rd\_set:

```
RP/0/0/CPU0:router(config)# rd-set my_rd_set
RP/0/0/CPU0:router(config-rd)# 172.16.0.0/16:*,
RP/0/0/CPU0:router(config-rd)# 172.17.0.0/16:100,
RP/0/0/CPU0:router(config-rd)# 192:*,
RP/0/0/CPU0:router(config-rd)# 192:100
RP/0/0/CPU0:router(config-rd)# end-set
```

# replace as-path

To replace a sequence of AS numbers or private AS numbers in the AS path with the configured local AS number, use the **replace as-path** command in route-policy configuration mode.

**replace as-path** {[*as-number-list parameter*]} **private-as**

## Syntax Description

<i>as-number-list</i>	(Optional) Sequence of AS numbers to replace. The sequence must be enclosed in single quotes ( ' '). You can use 2-byte or 4-byte AS numbers. <ul style="list-style-type: none"> <li>• The 2-byte value is entered as a 16-bit unsigned decimal value. The range is 0 to 65535.</li> <li>• The 4-byte value is entered as two 16-bit unsigned decimal values separated by a period. The range is 1.0 to 65535.65535.</li> </ul>
<i>parameter</i>	(Optional) Parameter name. The parameter name must be preceded with a "\$."
<b>private-as</b>	Matches within the BGP private AS range. Range is from 64512 to 65534.

## Command Default

None.

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **replace as-path** command to replace a sequence of AS numbers or private AS numbers in the AS path with the local AS numbers. For example, if the AS path is '67 65534 100 65533 5 78 89 90' and the local AS number is 900, then:

```
replace as-path '5 78'
```

replaces '5 78' in the AS path with 900 (from the local AS), and the new path would be '67 65534 100 65533 900 89 90'.

Consider following statement:

```
replace as-path private-as
```

Because 65534 and 65533 are within the private AS range, they are replaced with 900. The path is '67 900 100 900 5 78 89 90'. The length of the path remains the same.

The **replace as-path** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.



#### Caution

The **replace as-path** command changes the AS path content which can lead to routing loops.

#### Task ID

Task ID	Operations
route-policy	read, write

#### Examples

The following example shows how to use the **replace as-path** command to replace AS numbers in the AS path:

```
RP/0/0/CPU0:router(config)# route-policy drop-as-1234
RP/0/0/CPU0:router(config-rpl)# replace as-path '90 78 45 $asnum'
RP/0/0/CPU0:router(config-rpl)# replace as-path private-as
RP/0/0/CPU0:router(config-rpl)# replace as-path '9.9 7.89 14.15 $asnum'
RP/0/0/CPU0:router(config-rpl)# replace as-path '9 89 14.15 $asnum'
```

# rib-has-route

To check if a route listed in the prefix set exists in the Routing Information Base (RIB), use the **rib-has-route** command in route-policy configuration mode.

**rib-has-route in** {*prefix-set-name*|*inline-prefix-set*|*parameter*}

## Syntax Description

<i>prefix-set-name</i>	Name of a prefix set.
<i>inline-prefix-set</i>	Inline prefix set. The inline prefix set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>name</i> argument was removed. The following arguments were added: <ul style="list-style-type: none"><li>• <i>prefix-set-name</i></li><li>• <i>inline-prefix-set</i></li><li>• <i>parameter</i></li></ul>

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

If routes are active, then they are advertised. Routes are considered active if they are already installed in the Routing Information Base (RIB).

The prefix sets used in the **rib-has-route** command contain two match specifications. The first is where an exact route match is requested (for example, 10.10.0.0/16 will match exactly one route) and the second is where a route match or any more-specific route match is allowed (for example, 10.10.0.0/16 le 32 will match the 10.10.0.0/16 route and any longer prefix).

Use the **rib-has-route** command as a conditional expression within an **if** statement to check if there is an active route with a specific prefix contained in the RIB. If the statement reveals an active route that meets that criteria, additional actions are executed.

For a list of all conditional expressions available within an **if** statement, see the **if** command.

### Task ID

Task ID	Operations
route-policy	read, write

### Examples

In the following example, an **if** statement is used to learn if a route contained in a prefix set 10.10.0.0/16 is in the RIB:

```
RP/0/0/CPU0:router(config-rpl)# if rib-has-route in (10.10.0.0/16 ge 16) then  
RP/0/0/CPU0:router(config-rpl-if)# pass  
RP/0/0/CPU0:router(config-rpl-if)# endif  
RP/0/0/CPU0:router(config-rpl)#
```

# route-has-label

To check if there is a Multiprotocol Label Switching (MPLS) label in a route during redistribution, use the **route-has-label** command in route-policy configuration mode.

**route-has-label**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Command History	Release	Modification
	Release 3.3.0	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **route-has-label** command as a conditional expression within an **if** statement to check if there is an MPLS label in a route during redistribution.

For a list of all conditional expressions available within an **if** statement, see the **if** command.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** In the following example, an **if** statement learns if an MPLS label is present in a route:

```
RP/0/0/CPU0:router(config-rpl)# if route-has-label then
RP/0/0/CPU0:router(config-rpl-if)# pass
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

# route-policy (RPL)

To define a route policy and enter route-policy configuration mode, use the **route-policy** command in global configuration mode. To remove a policy definition, use the **no** form of this command.

**route-policy** *name* [(*parameter1*, *parameter2*, . . . , *parameterN*)]

**no route-policy** *name* (*parameter1*, *parameter2*, . . . , *parameterN*)

## Syntax Description

<i>name</i>	Name of a route policy.
<i>parameter</i>	(Optional) Parameter name. The parameter name must be preceded with a "\$." The <i>parameters</i> must be enclosed in parenthesis "()".

## Command Default

No default behavior or values

## Command Modes

Global configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **route-policy** command to define a route policy and enter route-policy configuration mode.

Policy definitions create named bundles of policy statements. A policy definition consists of the **route-policy** command followed by a name, a group of policy statements, and the **end-policy** command.

The policy name serves as a handle for binding the policy to protocols.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following example shows a simple policy named drop-everything that drops any route it encounters:

```
RP/0/0/CPU0:router(config)# route-policy drop-everything
RP/0/0/CPU0:router(config-rpl)# drop
```



```
RP/0/0/CPU0:router(config-rpl)# end-policy
```

Policies may also refer to other policies such that common blocks of policy can be reused. This reference to other policies is accomplished by using the **apply** command. The following is a simple example:

```
RP/0/0/CPU0:router(config)# route-policy drop-as-1234
RP/0/0/CPU0:router(config-rpl)# if as-path passes-through '1234' then
RP/0/0/CPU0:router(config-rpl-if)# apply check-communities
RP/0/0/CPU0:router(config-rpl-if)# else
RP/0/0/CPU0:router(config-rpl-else)# pass
RP/0/0/CPU0:router(config-rpl-else)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

The **apply** command indicates that the policy check-communities should be executed if the route under consideration passed through autonomous system 1234 before it was received. If so, the communities of the route are checked, and based on the findings the route may be accepted unmodified, accepted with changes, or dropped.

#### Related Commands

Command	Description
<a href="#">end-policy</a> , on page 55	Ends the definition of a route policy.

## route-type is

To match route types when redistribution is being performed into Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), or Integrated Intermediate System-to-Intermediate System (IS-IS), use the **route-type is** command in route-policy configuration mode.

**route-type is** {**local**| **interarea**| **internal**| **type-1**| **type-2**| **level-1**| **level-2**| *parameter*}

### Syntax Description

<b>local</b>	Uses a local value to match locally generated BGP routes.
<b>interarea</b>	Uses an interarea value to match IS-IS interarea routes.
<b>internal</b>	Uses an internal value to match OSPF intra- and interarea routes.
<b>type-1</b>	Uses a Type 1 value to match Type 1 OSPF routes.
<b>type-2</b>	Uses a Type 2 value to match Type 2 OSPF routes.
<b>level-1</b>	Uses a Level 1 value to match Level 1 IS-IS routes.
<b>level-2</b>	Uses a Level 2 value to match Level 2 IS-IS routes.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>level-1-2</b> keyword was replaced with the <b>interarea</b> keyword.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **route-type is** command as a conditional expression within an **if** statement to compare route types when redistribution is being performed into BGP, OSPF, or IS-IS.

**Note**

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The valid keywords are **local**, **internal**, **interarea**, **type-1**, **type-2**, **level-1**, and **level-2**. A parameterized value that fills in one of these values may also be used. The **local** value is used to match locally generated BGP routes. The **internal** value is used to match OSPF intra- and interarea routes. The **type-1** and **type-2** values are used to match Type 1 and Type 2 OSPF external routes. The **level-1**, **level-2**, and **interarea** values are used to match IS-IS routes of those respective types.

Because the route type is a matching operator, it appears in conditional clauses of **if** and **then** statements.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, non-local routes are dropped:

```
RP/0/0/CPU0:router(config)# route-policy policy_A
RP/0/0/CPU0:router(config-rpl)# if route-type is local then
RP/0/0/CPU0:router(config-rpl-if)# pass
RP/0/0/CPU0:router(config-rpl-if)# else
RP/0/0/CPU0:router(config-rpl-else)# drop
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# rpl editor

To set the default routing policy language (RPL) editor, use the **rpl editor** command in global configuration mode.

**rpl editor** {**nano**|**emacs**|**vim**}

## Syntax Description

<b>nano</b>	Sets the default RPL editor to GNU nano.
<b>emacs</b>	Sets the default RPL editor to EMACS.
<b>vim</b>	Sets the default RPL editor to VIM.

## Command Default

The Nano editor is the default.

## Command Modes

Global configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
Release 3.5.0	The <b>nano</b> keyword was added and the GNU nano text editor became the default editor type.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the default RPL editor is set to Nano:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# rpl editor nano
```

In the following example, the default RPL editor is set to EMACS:

```
RP/0/0/CPU0:router# configure  
RP/0/0/CPU0:router(config)# rpl editor emacs
```

In the following example, the default RPL editor is set to VIM:

```
RP/0/0/CPU0:router# configure  
RP/0/0/CPU0:router(config)# rpl editor vim
```

# rpl maximum

To configure system limits on the routing policy subsystem, use the **rpl maximum** command in global configuration mode.

**rpl maximum** {*lines*|*policies*} *number*

## Syntax Description

<b>lines</b> <i>number</i>	Configures the number of lines of configuration limit. Range is from 1 to 131072.
<b>policies</b> <i>number</i>	Configures the number of policies limit. Range is from 1 to 5000.

## Command Default

**lines** *number* : 65536  
**policies** *numbers* : 3500

## Command Modes

global configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **rpl maximum** command to configure system limits on the routing policy subsystem. As such, **rpl maximum** configuration lines do not appear as statements within a routing policy. This command places resource limits on the routing policy subsystem. Use the **rpl maximum** command to configure the maximum number of lines of configuration and number of policies.

The number of lines of configuration includes the beginning and ending statements , for example, **route-policy** and **end-policy** . Each line of configuration for sets is also counted.

A line of configuration is counted only once; it is not counted each time it is used. Similarly, any multiple use of policy in an apply statement counts only as one policy.

A user can change the default values for lines and policies but cannot exceed the maximum value, nor can the value for lines and policies be configured lower than the number of lines or policies that are currently configured.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, the maximum number of RPL system limits are modified:

```
RP/0/0/CPU0:router# configure  
RP/0/0/CPU0:router(config)# rpl maximum lines 50  
RP/0/0/CPU0:router(config)# rpl maximum policies 6
```

**Related Commands**

Command	Description
<a href="#">show rpl maximum, on page 237</a>	Displays the maximum limits for lines of configuration and number of policies.

# set community

To set the Border Gateway Protocol (BGP) community attributes in a route, use the **set community** command in route-policy configuration mode.

**set community** {*community-set-name*|*inline-community-set*|*parameter*} [**additive**]

## Syntax Description

<i>community-set-name</i>	Community set name.
<i>inline-community-set</i>	Inline community set. The inline community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>additive</b>	(Optional) Adds communities to communities in the route.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set community** command to set the BGP community attribute.



### Note

The **set community** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Communities are 32-bit values carried in BGP routes. Each route may have zero or more communities in an unordered list.

Use this command to replace the communities in a route or add to them using the optional **additive** keyword.

As with the other community forms that support inline sets, either or both 16-bit portions of the community can be parameterized. Likewise, the names of the well-known communities **internet** (0:0), **no-advertise**



(65535:65281), **no-export** (65535:65282), and **local-AS** (65535:65283) can also be used. In an inline community set, each 16-bit portion can also be specified as the **peer-as** to express the AS number of the neighbor from which the route was received. If the neighbor AS employs a 4-byte ASN, the IANA-assigned 16-bit value 23456 (AS\_TRANS) is used as **peer-as** instead.

Without the **additive** keyword, any existing communities (other than the well-known communities) are removed and replaced with the given communities. The **additive** keyword specifies that all communities already present in the route be maintained and the list of communities be added to them.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

The following are incomplete configuration examples using the **set community** command:

```
RP/0/0/CPU0:router(config-rpl)# set community (10:24)
RP/0/0/CPU0:router(config-rpl)# set community (10:24, $as:24, $as:$tag)
RP/0/0/CPU0:router(config-rpl)# set community (10:24, internet) additive
RP/0/0/CPU0:router(config-rpl)# set community (10:24, $as:24) additive
RP/0/0/CPU0:router(config-rpl)# set community (10:24, peer-as:24) additive
```

## set core-tree

To set a Multicast Distribution Tree (MDT) type, use the **set core-tree** command in route-policy configuration mode.

**set core-tree** {gre-rosen| mldp-inband| mldp-partitioned-mp2mp| mldp-partitioned-p2mp| mldp-rosen| rsvp-te-partitioned-p2mp| *parameter*}

### Syntax Description

<b>gre-rosen</b>	Specifies the IP GRE Rosen core MDT type
<b>mldp-inband</b>	Specifies the MLDP InBand core MDT type
<b>mldp-partitioned-mp2mp</b>	Specifies the MLDP Partitioned MP2MP core MDT type
<b>mldp-partitioned-p2mp</b>	Specifies the MLDP Partitioned P2MP core MDT type
<b>mldp-rosen</b>	Specifies the MLDP Rosen core MDT type
<b>rsvp-te-partitioned-p2mp</b>	Specifies the RSVP TE core core MDT type
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

None

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 4.1.0	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

### Task ID

Task ID	Operation
route-policy	read, write

## Examples

In this example, the Multicast Distribution Tree type is set to IP GRE Rosen core:

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#route-policy policy_mdt_type
RP/0/0/CPU0:router(config-rpl)#set core-tree gre-rosen
```

# set dampening

To configure Border Gateway Protocol (BGP) route dampening, use the **set dampening** command in route-policy configuration mode.

**set dampening** {**half-life** {*minutes*| *parameter*}| **max-suppress** {*minutes*| *parameter*}| **reuse** {*seconds*| *parameter*}| **suppress** {*penalty-units*| *parameter*}| **others default**}

## Syntax Description

<b>half-life</b> <i>minutes</i>	Specifies the time (in minutes) after which a penalty is decreased. After the route has been assigned a penalty, the penalty is decreased by half after the half-life period. The process of reducing the penalty happens every 5 seconds. Range is 1 to 45 minutes.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>max-suppress</b> <i>minutes</i>	Specifies the maximum time (in minutes) a route can be suppressed. Range is 1 to 20000. If the half-life value is allowed to default, the maximum suppress time defaults to 60 minutes.
<b>reuse</b> <i>seconds</i>	Unsuppresses a route if the penalty for flapping the route decreases enough to fall below the configured value (in seconds). The process of unsuppressing routes occurs at 10-second increments. Range is 1 to 20000.
<b>suppress</b> <i>penalty-units</i>	Specifies a penalty of 1000 each time a route flaps. When a route penalty exceeds the configured limit, it is suppressed. Range is 1 to 20000.
<b>others default</b>	If all four keyword values are not specified in the command, then the command <i>must</i> end with <b>others default</b> . This designation indicates that any keyword not defined is set to its default.

## Command Default

**half-life** : 15 minutes  
**max-suppress** : 60 minutes (four times the half-life)  
**reuse** : 750 seconds  
**suppress** : 2000 penalty units

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was supported.

**Usage Guidelines**

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

The BGP protocol supports route dampening using an exponential backoff algorithm. The algorithm is controlled by setting the four supported BGP values: half-life, max-suppress, reuse, and suppress. Use the **set dampening** command to configure BGP route dampening.

**Note**

The **set dampening** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

A value for at least one of the four keywords must be set. If the **set dampening** command defines values for three or fewer of the supported keywords, then the configuration must end with the **others default**, which indicates that any keyword value not defined in the command is set to its default value.

The keywords may appear in the command in any order.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following examples, the half-life is set to 20 minutes and the maximum suppress time is set to 90 minutes. Each command must end with **others default** because three or fewer keywords are defined.

```
RP/0/0/CPU0:router(config-rpl)# set dampening halflife 20 others default
RP/0/0/CPU0:router(config-rpl)# set dampening max-suppress 90 others default
```

In this example, all four keywords are defined, which means the command does not use **others default**.

```
RP/0/0/CPU0:router(config-rpl)# set dampening halflife 15 max-suppress 60 reuse 750 suppress 2000
```

The following command is invalid because it is missing **others default**.

```
RP/0/0/CPU0:router(config-rpl)# set dampening reuse 700
```

In the following example, the parameters are used.

```
RP/0/0/CPU0:router(config-rpl)# set dampening halflife $p1 suppress $p4 reuse $p3 max-suppress $p2
```

# set eigrp-metric

To set Enhanced Interior Gateway Routing Protocol (EIGRP) route metrics, use the **set eigrp-metric** command in route-policy configuration mode.

**set eigrp-metric** *bandwidth delay reliability loading mtu*

## Syntax Description

<i>bandwidth</i>	Minimum bandwidth of the route in kilobits per second. Range is 1 to 4294967295.
<i>delay</i>	Route delay in tens of microseconds. Delay is 1 or any positive number that is a multiple of 39.1 nanoseconds. Range is 1 to 4294967295.
<i>reliability</i>	Likelihood of successful packet transmission expressed as a number between 0 and 255. The value 255 means 100 percent reliability; 0 means no reliability.
<i>loading</i>	Effective bandwidth of the route expressed as a number from 1 to 255 (255 is 100 percent loading).
<i>mtu</i>	Minimum maximum transmission unit (MTU) size of the route in bytes. Range is from 1 to 65535.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

You can use the **add** command to further offset an existing EIGRP metric value.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the EIGRP metrics are adjusted for route policy policy\_1:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set eigrp-metric 1400 120 250 100 1500
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">add</a> , <a href="#">on page 8</a>	Adds an offset to an existing value.

## set extcommunity cost

To set the Border Gateway Protocol (BGP) cost extended community attributes, use the **set extcommunity cost** command in route-policy configuration mode.

**set extcommunity cost** {*cost-extcommunity-set-name*| *cost-inline-extcommunity-set*| *parameter*} [**additive**]

### Syntax Description

<i>cost-extcommunity-set-name</i>	Cost extended community set name.
<i>cost-inline-extcommunity-set</i>	Inline cost extended community set. The inline cost extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>additive</b>	(Optional) Adds extended communities for cost to extended communities in the route.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>set extcommunity</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set extcommunity cost** command to either replace the extended communities on the route or add to them using the optional **additive** keyword. Cost community is an extended community used to tie break the best path selection process in BGP so as to have a localized custom decision for packet forwarding. The extended community format defines generic points of insertion (POI) that influence the decision at different points of the bestpath algorithm.



**Note**

The **set extcommunity cost** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

As with the other extended community forms that support inline sets, either or both portions of the community can be parameterized. Similarly to regular communities, the **additive** keyword can be used to signify adding these extended communities to those that are already present, as opposed to replacing them. Without the **additive** keyword, any existing extended communities for cost (other than the well-known communities) are removed and replaced with the given communities. The **additive** keyword specifies that all extended communities for cost already present in the route be maintained and the set of extended communities be added to them. Well-known communities include internet, local-AS, no-advertise, and no-export.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following are incomplete configuration examples using the **set extcommunity cost** command:

```
RP/0/0/CPU0:router(config-rpl)# set extcommunity cost (IGP:10:20)
RP/0/0/CPU0:router(config-rpl)# set extcommunity cost (Pre-Bestpath:33:44)
RP/0/0/CPU0:router(config-rpl)# set extcommunity cost (IGP:11:21)
```

## set extcommunity rt

To set the Border Gateway Protocol (BGP) route target (RT) extended community attributes, use the **set extcommunity rt** command in route-policy configuration mode.

**set extcommunity rt** {*rt-extcommunity-set-name*|*rt-inline-extcommunity-set*|*parameter*} **additive**

### Syntax Description

<i>rt-extcommunity-set-name</i>	Route target extended community set name.
<i>rt-inline-extcommunity-set</i>	Inline route target extended community set. The inline route target extended community set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>additive</b>	(Optional) Adds extended communities for an RT to extended communities in the route.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.3.0	This command was introduced.
	This command replaced the <b>set extcommunity</b> command.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set extcommunity rt** command to either replace the extended communities on the route or add to them using the optional **additive** keyword.



#### Note

The **set extcommunity rt** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

As with the other extended community forms that support inline sets, either or both portions of the community can be parameterized. Similarly to regular communities, the **additive** keyword can be used to signify adding these extended communities to those that are already present, as opposed to replacing them.

### Task ID

Task ID	Operations
route-policy	read, write

### Examples

The following are incomplete configuration examples using the **set extcommunity rt** command:

```
RP/0/0/CPU0:router(config-rpl)# set extcommunity rt (10:24)
RP/0/0/CPU0:router(config-rpl)# set extcommunity rt (10:24, $as:24, $as:$tag)
RP/0/0/CPU0:router(config-rpl)# set extcommunity rt (10:24, internet) additive
RP/0/0/CPU0:router(config-rpl)# set extcommunity rt (10:24, $as:24) additive
```

Without the **additive** keyword, any existing extended communities for cost (other than the well-known communities) are removed and replaced with the given communities. The **additive** keyword specifies that all extended communities for cost already present in the route be maintained and the list of extended communities be added to them.

# set ip-precedence

To set the IP precedence, use the **set ip-precedence** command in route-policy configuration mode.

**set ip-precedence** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value of the precedence. The precedence value can be a number from 0 to 7: <ul style="list-style-type: none"> <li>• 7 —network (set packets with network control precedence)</li> <li>• 6 —internet (set packets with internetwork control precedence)</li> <li>• 5 —critical (set packets with critical precedence)</li> <li>• 4 —flash-override (set packets with flash override precedence)</li> <li>• 3 —flash (set packets with flash precedence)</li> <li>• 2 —immediate (set packets with immediate precedence)</li> <li>• 1 —priority (set packets with priority precedence)</li> <li>• 0 —routine (set packets with routine precedence)</li> </ul>
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.6.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set ip-precedence** command to set the IP precedence to classify packets. This command is supported at the BGP table-policy attachpoint. Prefixes are marked for subsequent processing in the forwarding plane. After QoS Policy Propagation through Border Gateway Protocol (BGP) (QPPB) is enabled on an interface, corresponding traffic shaping and policing is completed using packet classification based on the IP precedence or QoS group ID. See *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco XR 12000 Series Router* for information on QPPB.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

This example shows how use **set ip-precedence** command:

```
RP/0/0/CPU0:router(config)# route-policy policy_1  
RP/0/0/CPU0:router(config-rpl)# set ip-precedence 3  
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# set isis-metric

To set the Intermediate System-to-Intermediate System (IS-IS) metric attribute value, use the **set is-is metric** command in route-policy configuration mode.

**set isis-metric** {*number*|*parameter*}

## Syntax Description

<i>number</i>	24-bit integer number. Range is from 0 to 16777215.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set isis-metric** command to set the IS-IS metric attribute value for routes that are redistributed into IS-IS.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the IS-IS metric attribute value is set to 1000:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set isis-metric 1000
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# set label

To set the Border Gateway Protocol (BGP) label attribute value, use the **set label** command in route-policy configuration mode.

**set label** {**explicit-null**|**implicit-null**| *parameter*}

## Syntax Description

<b>explicit-null</b>	Sets the label to the well-known explicit value of 0.
<b>implicit-null</b>	Sets the label to the well-known implicit value of 3.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set label** command in a route policy at the allocate label attachpoint to set the label to explicit-null or implicit-null based on deployment preference. During inter-AS operation, the ASBR sends some of its own loopbacks to other its peers and labels them either implicit null or explicit null.

## Examples

The following example shows how to set the labels:

```
RP/0/0/CPU0:router(config)# route-policy labelpolicy
RP/0/0/CPU0:router(config-rpl)# if destination in (206.141.1.0/24) then
RP/0/0/CPU0:router(config-rpl)# set label explicit-null
RP/0/0/CPU0:router(config-rpl)# elseif destination in (206.141.3.0/24) then
RP/0/0/CPU0:router(config-rpl)# drop
RP/0/0/CPU0:router(config-rpl)# elseif destination in (206.141.4.0/24) then
RP/0/0/CPU0:router(config-rpl)# set label explicit-null
RP/0/0/CPU0:router(config-rpl)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## set label-mode

To set the type of Border Gateway Protocol (BGP) label mode, use the **set label-mode** command in route-policy configuration mode. This command does not have a **no** form.

**set label-mode** {per-ce | per-vrf | per-prefix}

### Syntax Description

<b>per-ce</b>	Specifies that the same label is used for all routes advertised from a unique customer edge (CE) peer or router.
<b>per-vrf</b>	Specifies that the same label is used for all routes advertised from a unique VRF.
<b>per-prefix</b>	Specifies that the same label is used for all routes advertised from a unique prefix.

### Command Default

Per-prefix label mode.

If a policy attached at label-mode attachpoint evaluates to pass and a **label mode** is not explicitly set, **per-prefix** is used as a default label mode.

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 4.3.1	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set label-mode** command in a route policy at the label-mode attachpoint to set the type of label mode to per-ce or per-vrf or per-prefix, based on deployment preference.

### Task ID

Task ID	Operation
route-policy	read, write

### Examples

This example shows how to set the type of label-mode to per-ce:

```
RP/0/0/CPU0:router(config)# route-policy set_label_mode
```



```
RP/0/0/CPU0:router(config-rpl)# set label-mode per-ce
RP/0/0/CPU0:router(config-rpl)# end-policy
```

This example shows how to set the type of label-mode to per-vrf:

```
RP/0/0/CPU0:router(config)# route-policy set_label_mode
RP/0/0/CPU0:router(config-rpl)# set label-mode per-vrf
RP/0/0/CPU0:router(config-rpl)# end-policy
```

This example shows how to set the type of label-mode to per-prefix:

```
RP/0/0/CPU0:router(config)# route-policy set_label_mode
RP/0/0/CPU0:router(config-rpl)# set label-mode per-prefix
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## Related Commands

Command	Description
<a href="#">route-policy (RPL), on page 120</a>	Defines a route policy and enters route-policy configuration mode.

# set level

To configure the Intermediate System-to-Intermediate System (IS-IS) link-state packet (LSP) level advertised to redistributed routes, use the **set level** command in route-policy configuration mode.

**set level** {**level-1**| **level-2**| **level-1-2**| *parameter*}

## Syntax Description

<b>level-1</b>	Specifies that redistributed routes are advertised in the Level 1 LSP of the router.
<b>level-2</b>	Specifies that redistributed routes are advertised in the Level 2 LSP of the router.
<b>level-1-2</b>	Specifies that redistributed routes are advertised in Level 1 and Level 2 LSPs of the router.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the IS-IS **set level** command to configure the LSP level advertised to redistributed routes.



### Note

The **set level** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

This command supports parameterization of the **level** keyword.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the level is set to Level 2:

```
RP/0/0/CPU0:router(config)# route-policy bgp_isis_redist
RP/0/0/CPU0:router(config-rpl)# if destination in (172.2.0.0/16 ge 16) then
RP/0/0/CPU0:router(config-rpl)# set level level-2
RP/0/0/CPU0:router(config-rpl)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# set local-preference

To set the Border Gateway Protocol (BGP) local preference attribute in a route, use the **set local-preference** command in route-policy configuration mode.

**set local-preference** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

Default value is 100.

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set local-preference** command to specify a preference value for the autonomous system path. Local preference is a nontransitive (does not cross autonomous system boundaries) attribute and is the second metric considered in the BGP best path calculation (the highest local preference is chosen). Weight is the first metric evaluated for best path, but it is local to the router and propagates only to iBGP peers. See the *Implementing BGP* module of the *Cisco IOS XR Routing Configuration Guide for the Cisco XR 12000 Series Router* for information on the BGP best path calculation.



### Note

The **set local-preference** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

The local preference is a 32-bit unsigned integer.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the local preference value is set to 10:

```
RP/0/0/CPU0:router(config-rpl)# set local-preference 10
```

# set med

To set the Border Gateway Protocol (BGP) Multi Exit Discriminator (MED) attribute, use the **set med** command in route-policy configuration mode.

**set med** {*number*|*parameter*} **igp-cost** {+| {*number*|*parameter*}|-| {*number*|*parameter*} }| **max-reachable**}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>igp-cost</b>	Sets the MED value to the cost for the Interior Gateway Protocol (IGP) route to resolve the next-hop of the BGP route.
+   -	Sets the MED to the MED plus or minus a static offset. An integer or parameter must follow the plus or minus.
<b>max-reachable</b>	Sets the MED value to the maximum possible value of 4294967295.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set med** command to set the MED value, which is a 32-bit unsigned integer.



### Note

The **set med** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

This command can take the following as argument values: an integer, a parameter, the **igp-cost** keyword, or a mathematical operator (either plus or minus) followed by an integer or a parameter. Setting the MED to the IGP cost is supported on outbound BGP policies only. The MED cannot be set to the IGP cost in policies applied to other BGP attach points.

The **max-reachable** keyword sets the MED to the maximum value while leaving the route reachable.

The plus or minus variants allow the user to set the MED to the MED plus or minus a static offset. The variants that allow a user to add or subtract offsets to the MED value are also range checked for underflow or overflow. If the value underflows as a result of subtraction, then the MED value is set to zero. If the value overflows, the value is set to 4294967295, which is the maximum value for MED. when MED is set to 4294967295, the route is unreachable.

#### Task ID

Task ID	Operations
route-policy	read, write

#### Examples

The following two examples show how to set the MED to a value that is either specified directly (using the integer 156) or passed to the policy as a parameter:

```
RP/0/0/CPU0:router(config-rpl)# set med 156
RP/0/0/CPU0:router(config-rpl)# set med $med_param
```

The following example shows how to instruct BGP to automatically set the MED value to the cost of the IGP route that resolves the next-hop of the BGP route:

```
RP/0/0/CPU0:router(config-rpl)# set med igp-cost
```

## set metric-type (IS-IS)

To configure the integrated Intermediate System-to-Intermediate System (IS-IS) metric type, use the **set metric-type** command in route-policy configuration mode.

**set metric-type** {**internal**| **external**| **rib-metric-as-internal**| **rib-metric-as-external**| *parameter*}

### Syntax Description

<b>internal</b>	Sets metric type to internal.
<b>external</b>	Sets the metric type to external.
<b>rib-metric-as-internal</b>	Uses RIB metric and sets IS-IS internal metric type.
<b>rib-metric-as-external</b>	Uses RIB metric and sets IS-IS external metric type.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.8.0	The <b>rib-metric-as-internal</b> and <b>rib-metric-as-external</b> keywords were added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the IS-IS **set metric-type** command to control whether IS-IS treats the metric as an internal or external metric.

Use the **rib-metric-as-external** and **rib-metric-as-internal** keywords to preserve RIB metrics when redistributing routes from another ISIS router instance or another protocol.



#### Note

The **set metric-type** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.



This command does not support parameterization.

### Task ID

Task ID	Operations
route-policy	read, write

### Examples

In the following example, the IS-IS metric type is set to internal:

```
RP/0/0/CPU0:router(config-rpl)# set metric-type internal
```

## set metric-type (OSPF)

To control how Open Shortest Path First (OSPF) computes the cost for a route, use the **set metric-type** command in route-policy configuration mode.

**set metric-type** {**type-1**| **type-2**| *parameter*}

### Syntax Description

<b>type-1</b>	Uses the cost set on the route plus the topology-related costs in the calculation for Type 1 metrics.
<b>type-2</b>	Uses only the cost set on the route in the calculation for Type 2 metrics.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the OSPF **set metric-type** command to control whether OSPF treats the cost as a Type 1 or Type 2 metric.



#### Note

The **set metric-type** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

The value of Type 1 or Type 2 controls how OSPF computes the cost for this route. For Type 2 metrics, only the cost set on the route is used. For Type 1 metrics, the cost set on the route plus the topology-related costs are used in the calculation.

This command does not support parameterization.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, the OSPF metric type is set to Type 1:

```
RP/0/0/CPU0:router(config-rpl)# set metric-type type-1
```

## set next-hop

To replace the next-hop associated with a given route, use the **set next-hop** command in route-policy configuration mode.

**set next-hop** {*ipv4-address*|*ipv6-address*|**peer-address**|*parameter*|**self**} [**destination-vrf**]

### Syntax Description

<i>ipv4-address</i>	Valid IPv4 address.
<i>ipv6-address</i>	Valid IPv6 address.
<b>peer-address</b>	Sets the next-hop to the IP address of the remote Border Gateway Protocol (BGP) peer.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>self</b>	Sets itself as the next-hop.
<b>destination-vrf</b>	(Optional) Specifies that the next-hop of the route should be resolved in destination VRF context. This keyword is available when an IPv4 or IPv6 address or parameter is used.

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>self</b> keyword was added.
Release 3.4.0	The <b>destination-vrf</b> keyword was added.

### Usage Guidelines

Use the **set next-hop** command to replace the next-hop associated with a specific address.



#### Note

The **set next-hop** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Use the **set next-hop peer-address** command to set the next-hop to the address of the BGP neighbor, where this policy is attached.

The next-hop is a valid IPv4 address entered as a dotted decimal or an IPv6 address entered as a colon-separated hexadecimal.

It is not possible to use this command to set the BGP IPv6 link-local next-hop.

The **destination-vrf** keyword is used mainly in Layer 3 VPN networks when importing routes.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the next-hop is set to a valid IPv4 address:

```
RP/0/0/CPU0:router(config-rpl) # set next-hop 10.0.0.5
```

In this example, the next-hop is set to a parameter value \$nexthop:

```
RP/0/0/CPU0:router(config-rpl) # set next-hop $nexthop
```

In this example, the next-hop is set to a valid IPv4 address with a destination VRF context:

```
RP/0/0/CPU0:router(config-rpl) # set next-hop 10.0.0.5 destination-vrf
```

# set origin

To change the Border Gateway Protocol (BGP) origin attribute, use the **set origin** command in route-policy configuration mode.

**set origin** {**igp**|**incomplete**|**egp**} *parameter*

## Syntax Description

<b>igp</b>	Sets the origin type to Interior Gateway Protocol (IGP).
<b>incomplete</b>	Sets the origin type to incomplete.
<b>egp</b>	Sets the origin type to Exterior Gateway Protocol (EGP).
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set origin** command to change the origin attribute.



### Note

The **set origin** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

The origin of a Border Gateway Protocol (BGP) route is **igp** , **egp** , or **incomplete** .

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the origin attribute is set to EGP:

```
RP/0/0/CPU0:router(config-rpl)# set origin egp
```

# set ospf-metric

To set an Open Shortest Path First (OSPF) protocol metric attribute value, use the **set ospf-metric** command in route-policy configuration mode.

**set ospf-metric** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 24-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set ospf-metric** command to set the metric for routes that are redistributed into OSPF. The OSPF metric operator accepts either an integer value or a parameter.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the OSPF metric attribute value is set to 1000:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set ospf-metric 1000
RP/0/0/CPU0:router(config-rpl)# end-policy
```



## set qos-group (RPL)

To set the quality of service (QoS) group, use the **set qos-group** command in route-policy configuration mode:

```
set qos-group {number|parameter}
```

### Syntax Description

<i>number</i>	QoS group ID. Range is from 0 to 31.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.6.0	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set qos-group** command to set the QoS group to classify packets.

This command is supported at the BGP table-policy attachpoint. Prefixes are marked for subsequent processing in the forwarding plane. After QoS Policy Propagation through Border Gateway Protocol (BGP) (QPPB) is enabled on an interface, corresponding traffic shaping and policing is completed using packet classification based on the IP precedence or QoS group ID. See the *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco XR 12000 Series Router* for information on QPPB.

### Task ID

Task ID	Operations
route-policy	read, write

### Examples

This example shows how to use **set qos-group** command:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set qos-group 12
```

```
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# set rib-metric

To set the Routing Information Base (RIB) metric attribute value for a table policy, use the **set rib-metric** command in route-policy configuration mode:

**set rib-metric** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set rib-metric** command set the RIB metric attribute value for BGP routes.

Every route in the RIB has a metric associated with it, signifying the cost to reach a specific destination based on link characteristics. The **set rib-metric** command modifies the RIB metric while installing BGP routes into RIB, enabling the upgrading or downgrading of the BGP route installed in RIB.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the RIB metric attribute is set to 1000:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set rib-metric 1000
RP/0/0/CPU0:router(config-rpl)# end-policy
```

# set rip-metric

To set Routing Information Protocol (RIP) metric attributes, use the **set rip-metric** command in route-policy configuration mode.

**set rip-metric** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 4-bit unsigned integer. Range is from 0 to 16.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.
Release 3.4.0	The <i>number</i> range of 0 to 15 was changed to 0 to 16.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set rip-metric** command to set the cost attribute for routes that are redistributed into RIP.

You can use the **add** command to increment the RIP metric value.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the RIP metric number is adjusted for route policy policy\_1:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set rip-metric 10
RP/0/0/CPU0:router(config-rpl)# end-policy
```

**Related Commands**

Command	Description
<a href="#">add</a> , on page 8	Adds an offset to an existing value.

# set rip-tag

To set a route tag attribute for Routing Information Protocol (RIP) routes, use the **set rip-tag** command in route-policy configuration mode.

**set rip-tag** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 16-bit unsigned integer. Range is from 0 to 65535.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set rip-tag** command to set the RIP tag attribute for routes that are redistributed into RIP. The RIP tag operator accepts either an integer value or a parameter.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the RIP tag is adjusted for route policy policy\_1:

```
RP/0/0/CPU0:router(config)# route-policy policy_1
RP/0/0/CPU0:router(config-rpl)# set rip-tag 1000
RP/0/0/CPU0:router(config-rpl)# end-policy
```

## set rpf-topology

To set reverse-path forwarding (RPF) to any default or nondefault tables for particular sources and groups, use the **set rpf-topology** command in routing policy configuration mode.

**set rpf-topology** [*vrf vrf-name*] {*ipv4*|*ipv6*} {*unicast*|*multicast*|*parameter*} **topology** *table-name*

### Syntax Description

<b>vrf</b> <i>vrf-name</i>	[Optional] Specifies a VPN routing and forwarding (VRF) instance. Required when configuring extranet topologies
<b>ipv4</b>	[Optional] Specifies IPv4 address prefixes.
<b>ipv6</b>	[Optional] Specifies IPv6 address prefixes.
<b>unicast</b>	Specifies unicast address prefixes.
<b>multicast</b>	Specifies multicast address prefixes.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>topology</b>	Specifies the default or non-default topology table for the source or group.
<i>table-name</i>	Alphanumeric name string.

### Command Default

Default or current topology setting.

### Command Modes

Routing policy configuration

### Command History

Release	Modification
Release 3.7.0	This command was introduced.
Release 3.8.0	This command was supported for MVPN extranet routing configuration. The <b>vrf</b> <i>vrf-name</i> keyword and argument were added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

When using this command for MVPN extranet routing configuration, only the **vrf** *vrf-name* keyword and argument are required.

When using this command in the context of multitopology routing, all keywords and arguments with the exception of **vrf** *vrf-name* keyword and argument are required.

## Task ID

Task ID	Operations
multicast	read

## Examples

The following example shows how to execute the **set rpf-topology** command:

```
RP/0/0/CPU0:router# config
RP/0/0/CPU0:router(config)# route-policy green
RP/0/0/CPU0:router(config-rpl)# set rpf-topology ipv6 multicast topology t12
```

The following example shows the use of **set rpf-topology** command in the context of creating an RPF for a topology table in multiple topologies:

```
route-policy mt4-p1
  if destination in (225.0.0.1, 225.0.0.11) then
    set rpf-topology ipv4 multicast topology t201
  elseif destination in (225.0.0.2, 225.0.0.12) then
    set rpf-topology ipv4 multicast topology t202
  elseif destination in (225.0.0.3, 225.0.0.13) then
    pass
  endif
end-policy
!

route-policy mt4-p3
  if destination in (225.0.0.8) then
    set rpf-topology ipv4 multicast topology t208
  elseif destination in (225.0.0.9) then
    set rpf-topology ipv4 multicast topology t209
  elseif destination in (225.0.0.10) then
    set rpf-topology ipv4 multicast topology t210
  else
    drop
  endif
end-policy
!
```

## Related Commands

Command	Description
rpf topology	Assigns a route policy in PIM to select a reverse-path forwarding (RPF) topology.



## set rtset route-limit

To set limits on paths in the inbound neighbor policy in BGP, particularly when acting as a route-reflector, use the **set rtset route-limit** command in the global configuration mode. If such a path is accepted, BGP adds a flag to the path, BPATH\_RTSET\_NET\_COUNT, to indicate that the path is subjected to the limit.

To remove a limit, use the **no** form of this command.

### set rtset route-limit

Syntax Description	limit-value	Displays the 32-unit quantity.
--------------------	-------------	--------------------------------

Command Default	No default behavior or values.
-----------------	--------------------------------

Command Modes	Global configuration
---------------	----------------------

Command History	Release	Modification
	Release 5.0	This command was introduced.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Task ID	Task ID	Operations
	route-policy	read

Examples	Based on service agreement, if a user AVPN advertises VPN prefixes with RT-set {1:1, 2:2} and user Team10 advertises VPN prefixes with RTs {3:3, 4:4}. On the DUT, the user wants to limit AVPN prefixes to 100, Team10 prefixes to 50, and for all other users, limit each of their prefixes to 80. Note that the fallback limit is per user, not a cumulative one. To achieve this, the user must configure the following route-policy:
----------	---

```
route-policy RTSET-LIMIT
  if extcommunity rt matches-every (1:1, 2:2) then
    set rtset-route-limit 100
  elseif extcommunity rt matches-every (3:3, 4:4) then
    set reset-route-limit 50
  else
    set reset-route-limit 80
  endif
end-policy
```

With this configuration, the RR will keep

- (i) at most 100 prefixes that have a path with RT-set containing {1:1, 2:2},
- (ii) at most 50 prefixes that have a path with RT-set containing {3:3, 4:4},  
and
- (iii) at most 80 prefixes that only have paths with RT-set that do not  
contain either {1:1, 2:2} or {3:3, 4:4}

## set spf-priority

To set OSPF Shortest Path First (SPF) priority, use the set spf-priority command in route-policy configuration mode.

**set spf-priority** {critical| high| medium}

### Syntax Description

<b>critical</b>	Sets critical priority for SPF
<b>high</b>	Sets high priority for SPF
<b>medium</b>	Sets medium priority for SPF

### Command Default

None

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.7.0	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

### Task ID

Task ID	Operation
route-policy	read, write

### Examples

This example sets SPF priority as critical:

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#route-policy policy_spf_priority
RP/0/0/CPU0:router(config-rpl)#set spf-priority critical
```

**Related Commands**

Command	Description
spf prefix-priority (OSPF)	Prioritizes OSPFv2 prefix installation into the global Routing Information Base (RIB) during Shortest Path First (SPF) run.

# set tag

To set the tag attribute, use the **set tag** command in route-policy configuration mode.

**set tag** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is from 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set tag** command to set the tag attribute.



### Note

The **set tag** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Tags are routing-protocol independent 32-bit integers that can be associated with a given route in the Routing Information Base (RIB).

For the Border Gateway Protocol (BGP), the tag attribute can be set only at the table-policy attach point.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the tag attribute is set to 10:

```
RP/0/0/CPU0:router(config-rpl)# set tag 10
```

In this example, the tag attribute is set to a parameter value \$tag\_param:

```
RP/0/0/CPU0:router(config-rpl)# set tag $tag_param
```

# set traffic-index

To set the traffic index attribute, use the **set traffic-index** command in route-policy configuration mode.

**set traffic-index** {*number*|*parameter*|**ignore**}

## Syntax Description

<i>number</i>	Integer value assigned to the traffic index attribute. Range is 1 to 63.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."
<b>ignore</b>	Specifies that Border Gateway Protocol (BGP) policy accounting is not done.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set traffic-index** command to set the traffic index attribute.



### Note

The **set traffic-index** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Traffic index is a special attribute for BGP. It is used as an index to a set of counters that are maintained by forwarding hardware. It is also used to track packet and byte counters that are forwarded using routes with specific attributes. These counters can be enabled and disabled on an individual interface basis.

The traffic index attribute can be set only at the table-policy attach point, and can take a value from 1 to 63, or a value of **ignore**. If the traffic index is set to **ignore**, then BGP policy accounting is not done. Parameterization of this value is also supported.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, a policy is created in which the traffic index is set to 10 for all routes that originated in autonomous system 1234:

```
RP/0/RSP0RP0/CPU0:router(config)# route-policy count-as-1234
RP/0/RSP0RP0/CPU0:router(config-rpl)# if as-path originates-from '1234' then
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# set traffic-index 10
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# else
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# pass
RP/0/RSP0RP0/CPU0:router(config-rpl-if)# endif
RP/0/RSP0RP0/CPU0:router(config-rpl)# end-policy
```

This policy could then be attached using the BGP **table-policy** command. The counters could then be enabled on various interfaces with the appropriate commands.



# set vpn-distinguisher

To change the Border Gateway Protocol (BGP) VPN distinguisher attribute, use the **set vpn-distinguisher** command in route-policy configuration mode.

**set vpn-distinguisher** *{number|parameter}*

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is from 1 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set vpn-distinguisher** command to change the VPN distinguisher attribute.



### Note

The **set origin** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

A VPN distinguisher is used in Layer 3 VPN networks for enhanced individual VPN control and to avoid route target mapping at AS boundaries in inter-AS VPN networks. Route target extended communities are removed at neighbor outbound, and the VPN distinguisher value is applied on the BGP route as an extended community. When the route is received on a neighboring router in another AS, the VPN distinguisher is removed and mapped to a route target extended community.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the VPN distinguisher attribute is set to 456:

```
RP/0/0/CPU0:router(config-rpl)# set vpn-distinguisher 456
```

# set weight

To set the weight value for Border Gateway Protocol (BGP) routes, use the **set weight** command in route-policy configuration mode.

**set weight** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Number assigned to the weight value for BGP routes. Weight is 16 bits. Range is 0 to 65535.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **set weight** command to set the weight value for BGP routes.



### Note

The **set weight** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

A weight is a value that can be applied to a route to override the BGP local preference. This is not a BGP attribute announced to BGP peer routers. RPL can be used to set the weight value.

Given two BGP routes with the same network layer reachability information (NLRI), a route with a higher weight is selected, no matter what the values of other BGP attributes may be. However, weight only has significance on the local router. It is not sent from one BGP speaker to another, even within the same autonomous system.

On Cisco routers, if a BGP route is sourced by the local router, its weight is automatically set to 32768; if the BGP route is learned from another router, its weight is automatically set to 0. Thus, by default, locally sourced routes are preferred over BGP learned routes.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, the weight of the route is set to 10 and then to a parameter value \$weight\_param:

```
RP/0/0/CPU0:router(config-rpl)# set weight 10
RP/0/0/CPU0:router(config-rpl)# set weight $weight_param
```

# show rpl

To display system-wide RPL configuration, use the **show rpl** command in EXEC mode.

**show** [**running-config**] **rpl** [**maximum** {*lines configuration-limit*|**policies** *policies-limit*}|**editor** {**emacs**|**nano**|**vim**}]

## Syntax Description

<b>running-config</b>	(Optional) Displays configuration-limit argument.
<b>maximum</b>	(Optional) Displays the maximum number of lines of configuration and number of policies.
<b>lines</b> <i>configuration-limit</i>	(Optional) Displays the number of lines to which configuration is limited. Range is 1 to 131072.  The <i>configuration-limit</i> argument is available if the <b>running-config</b> keyword is specified.
<b>policies</b> <i>policies-limit</i>	(Optional) Displays the limit on the number of policies. Range is 1 to 5000.  The <i>configuration-limit</i> argument is available if the <b>running-config</b> keyword is specified.
<b>editor</b>	(Optional) Specifies the default RPL editor. This keyword is available if the <b>running-config</b> keyword is specified.
<b>emacs</b>	(Optional) Displays the default RPL editor to Micro Emacs.
<b>nano</b>	(Optional) Displays the default RPL editor to nano.
<b>vim</b>	(Optional) Displays the default RPL editor to Vim.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

The following shows the output of the **show running-config rpl** command:

```
RP/0/0/CPU0:router# show running-config rpl

extcommunity-set rt ext_comm_set_rt_ex1
  1.2.3.4:34
end-set
!
prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any com_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
```

**Related Commands**

Command	Description
<a href="#">show rpl maximum, on page 237</a>	Displays the maximum limits for lines of configuration and number of policies.

## show rpl active as-path-set

To display the AS path sets that are referenced by at least one policy that is being used at an attach point, use the **show rpl active as-path-set** command in EXEC mode.

### show rpl active as-path-set [detail]

#### Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for active AS path sets.
---------------	---

#### Command Default

No default behavior or values

#### Command Modes

EXEC

#### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	Added the <b>detail</b> keyword.

#### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl active as-path-set** command to display all AS path sets that are in use in the system and referenced either directly or indirectly at a policy attach point.

#### Task ID

Task ID	Operations
route-policy	read

#### Examples

This example shows the following sample configuration:

```
router bgp 2
  address-family ipv4 unicast
  !
  neighbor 10.0.101.2
    remote-as 100
    address-family ipv4 unicast
    route-policy policy_1 in
  !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
RP/0/0/CPU0:router# show rpl route-policy policy_2 detail

prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!
community-set comm_set_ex1
  65500:1,
  65500:2,
  65500:3
end-set
!
extcommunity-set rt ext_comm_set_rt_ex1
  1.2.3.4:34
end-set
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!

RP/0/0/CPU0:router# show rpl route-policy policy_1 detail

prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!
as-path-set as_path_set_ex1
  ios-regex '^_655--$',
  ios-regex '^_65501_$'
end-set
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!

```

Given this sample configuration, the **show rpl active as-path-set** command displays the following information:

```

RP/0/0/CPU0:router# show rpl active as-path-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following as-path-sets are ACTIVE
-----
as_path_set_ex1

```



**Related Commands**

Command	Description
<a href="#">show rpl active community-set, on page 186</a>	Displays the community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active extcommunity-set, on page 189</a>	Displays the extended community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the route policies that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the prefix sets that are referenced by at least one policy that is being used at an attach point.

# show rpl active community-set

To display the community sets that are referenced by at least one policy that is being used at an attach point, use the **show rpl active community-set** command in EXEC mode.

**show rpl active community-set [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for active community sets.
---------------	---

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl active community-set** command to display all community sets that are in use in the system and referenced either directly or indirectly at a policy attach point.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
 remote-as 100
 address-family ipv4 unicast
 route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!

RP/0/0/CPU0:router# show rpl route-policy policy_2 detail

prefix-set prefix_set_ex1
 10.0.0.0/16 ge 16 le 32,
 0.0.0.0/0 ge 25 le 32,
 0.0.0.0/0
end-set
!
community-set comm_set_ex1
 65500:1,
 65500:2,
 65500:3
end-set
!
extcommunity-set rt_ext_comm_set_rt_ex1
 1.2.3.4:34
end-set
!

route-policy policy_2
 if destination in prefix_set_ex1 then
  if (community matches-any comm_set_ex1) then
   set community (10:666) additive
  endif
  if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
   set community (10:999) additive
  endif
 endif
end-policy
!

RP/0/0/CPU0:router# show rpl route-policy policy_1 detail

prefix-set prefix_set_ex1
 10.0.0.0/16 ge 16 le 32,
 0.0.0.0/0 ge 25 le 32,
 0.0.0.0/0
end-set
!
as-path-set as_path_set_ex1
 ios-regex '^_655--$',
 ios-regex '^_65501_$'
end-set
!
route-policy policy_1
 if (destination in prefix_set_ex1) then
  set local-preference 100
 endif
 if (as-path in as_path_set_ex1) then
  set community (10:333) additive
 endif
end-policy
!

```

Given this sample configuration, the **show rpl active community-set** command displays the following information:

```

RP/0/0/CPU0:router# show rpl active community-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```
The following community-sets are ACTIVE
-----
comm_set_ex1
```

## Related Commands

Command	Description
<a href="#">show rpl active as-path-set, on page 183</a>	Displays the AS path sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active extcommunity-set, on page 189</a>	Displays the extended community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the route policies that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the prefix sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active rd-set, on page 195</a>	Displays the route distinguisher sets that are referenced by at least one policy that is being used at an attach point.

## show rpl active extcommunity-set

To display the extended community sets for cost, route target (RT), and Site-of-Origin (SoO) that are referenced by at least one route policy used at an attach point, use the **show rpl active extcommunity-set** command in EXEC mode.

**show rpl active extcommunity-set** [**cost**|**rt**|**soo**] [**detail**]

### Syntax Description

<b>cost</b>	(Optional) Displays all extended community cost sets.
<b>rt</b>	(Optional) Displays all extended community RT sets.
<b>soo</b>	(Optional) Displays all extended community SoO sets.
<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for active extended community sets.

### Command Default

All extended community sets are displayed.

### Command Modes

EXEC

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	Added the <b>cost</b> , <b>rt</b> , <b>soo</b> , and <b>detail</b> keywords.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl active extcommunity-set** command to display all extended community sets that are in use in the system and referenced either directly or indirectly at a policy attach point.

### Task ID

Task ID	Operations
route-policy	read

**Examples**

This example shows the following sample configuration:

```

router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
 !
 neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
   route-policy policy_2 in
 !
 !
 !

RP/0/0/CPU0:router# show rpl route-policy policy_2 detail

prefix-set prefix_set_ex1
 10.0.0.0/16 ge 16 le 32,
 0.0.0.0/0 ge 25 le 32,
 0.0.0.0/0
end-set
!
community-set comm_set_ex1
 65500:1,
 65500:2,
 65500:3
end-set
!
extcommunity-set rt_ext_comm_set_rt_ex1
 1.2.3.4:34
end-set
!

route-policy policy_2
 if destination in prefix_set_ex1 then
  if (community matches-any comm_set_ex1) then
   set community (10:666) additive
  endif
  if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
   set community (10:999) additive
  endif
 endif
end-policy
!

RP/0/0/CPU0:router# show rpl route-policy policy_1 detail

prefix-set prefix_set_ex1
 10.0.0.0/16 ge 16 le 32,
 0.0.0.0/0 ge 25 le 32,
 0.0.0.0/0
end-set
!
as-path-set as_path_set_ex1
 ios-regex '^_655--$',
 ios-regex '^_65501_$'
end-set
!
route-policy policy_1
 if (destination in prefix_set_ex1) then
  set local-preference 100
 endif
 if (as-path in as_path_set_ex1) then
  set community (10:333) additive

```

```

endif
end-policy
!

```

Given this sample configuration, the **show rpl active extcommunity-set** command displays the following information:

```

RP/0/0/CPU0:router# show rpl active extcommunity-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached

UNUSED -- Not attached (directly or indirectly) and not referenced

The following extcommunity-sets are ACTIVE
-----
ext_comm_set_rt_ex1

```

## Related Commands

Command	Description
<a href="#">show rpl active as-path-set, on page 183</a>	Displays the AS path sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active community-set, on page 186</a>	Displays the community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the route policies that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the prefix sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active rd-set, on page 195</a>	Displays the route distinguisher sets that are referenced by at least one policy that is being used at an attach point.

# show rpl active prefix-set

To display the prefix sets that are referenced by at least one policy that is being used at an attach point, use the **show rpl active prefix-set** command in EXEC mode.

**show rpl active prefix-set [detail]**

Syntax Description	detail
	(Optional) Displays the content of the object and all referenced objects for active prefix sets.

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.2	This command was introduced.
	Release 3.3.0	The <b>detail</b> keyword was added.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Use the **show rpl active prefix-set** command to display all prefix sets that are in use in the system and referenced either directly or indirectly at a policy attach point.

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
 remote-as 100
 address-family ipv4 unicast
 route-policy policy_1 in
 !
```



```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!

RP/0/0/CPU0:router# show rpl route-policy policy_2 detail

prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!

community-set comm_set_ex1
  65500:1,
  65500:2,
  65500:3
end-set
!
extcommunity-set rt_ext_comm_set_rt_ex1
  1.2.3.4:34
end-set
!

route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!

RP/0/0/CPU0:router# show rpl route-policy policy_1 detail

prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!
as-path-set as_path_set_ex1
  ios-regex '^_655--$',
  ios-regex '^_65501_$'
end-set
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!

```

The following example displays active prefix sets:

```

RP/0/0/CPU0:router# show rpl active prefix-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

**show rpl active prefix-set**

```
The following prefix-sets are ACTIVE
-----
prefix_set_1
```

**Related Commands**

Command	Description
<a href="#">show rpl active as-path-set, on page 183</a>	Displays the AS path sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active community-set, on page 186</a>	Displays the community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active extcommunity-set, on page 189</a>	Displays the extended community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl route-policy attachpoints, on page 259</a>	Displays the route policies that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active rd-set, on page 195</a>	Displays the route distinguisher sets that are referenced by at least one policy that is being used at an attach point.

# show rpl active rd-set

To display the route distinguisher (RD) sets that are referenced by at least one policy that is being used at an attach point, use the **show rpl active rd-set** command in EXEC mode.

**show rpl active rd-set [detail]**

Syntax Description	<b>detail</b> (Optional) Displays the content of the object and all referenced objects for active route policies.
--------------------	---

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.6.0	This command was introduced.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Use the **show rpl active rd-set** command to display all RD sets that are in use in the system and that are referenced either directly or indirectly at a policy attach point.

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
rd-set rdset1
  10:151,
  100.100.100.1:153,
  100.100.100.62/31:63
end-set
!
rd-set rdset2
  10:152,
  100.100.100.1:154,
  100.100.100.62/31:89
```

```

end-set
!
route-policy rdsetmatch
  if rd in rdset1 then
    set community (10:112)
  elseif rd in rdset2 then
    set community (10:223)
  endif
end-policy
!
router bgp 10
  bgp router-id 10.0.0.1
  address-family vpnv4 unicast
neighbor 10.10.10.1
  remote-as 10
  address-family ipv4 unicast
  route-policy rdsetmatch in
!
!

```

Given this sample configuration, the **show rpl active rd-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl active rd-set
```

```
ACTIVE -- Referenced by at least one policy which is attached INACTIVE -- Only referenced
by policies which are not attached UNUSED -- Not attached (directly or indirectly) and not
referenced
```

```
The following rd-sets are ACTIVE
```

```
-----
      rdset1
      rdset2
```

## Related Commands

Command	Description
<a href="#">show rpl active as-path-set, on page 183</a>	Displays the AS path sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active community-set, on page 186</a>	Displays the community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active extcommunity-set, on page 189</a>	Displays the extended community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the prefix sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the route policies that are referenced by at least one policy that is being used at an attach point.

# show rpl active route-policy

To display the route policies that are referenced by at least one policy that is being used at an attach point, use the **show rpl active route-policy** command in EXEC mode.

**show rpl active route-policy [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for active route policies.
---------------	---

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>policy</b> keyword was changed to <b>route-policy</b> .
Release 3.3.0	The <b>detail</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl active route-policy** command to display all policies that are in use in the system and that are referenced either directly or indirectly at a policy attach point.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
```

```

    route-policy policy_1 in
    !
    !
neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
    route-policy policy_2 in
    !
    !
    !
RP/0/0/CPU0:router# show rpl route-policy policy_1

route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
RP/0/0/CPU0:router# show rpl route-policy policy_2

route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
```

Given this sample configuration, the **show rpl active route-policy** command displays the following information:

```

RP/0/0/CPU0:router# show rpl active route-policy

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following policies are (ACTIVE)
-----
policy_1
policy_2
```

## Related Commands

Command	Description
<a href="#">show rpl active as-path-set, on page 183</a>	Displays the AS path sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active community-set, on page 186</a>	Displays the community sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active extcommunity-set, on page 189</a>	Displays the extended community sets that are referenced by at least one policy that is being used at an attach point.

Command	Description
<a href="#">show rpl active prefix-set, on page 192</a>	Displays the prefix sets that are referenced by at least one policy that is being used at an attach point.
<a href="#">show rpl active rd-set, on page 195</a>	Displays the route distinguisher sets that are referenced by at least one policy that is being used at an attach point.

# show rpl as-path-set

To display the contents of AS path sets, use the **show rpl as-path-set** command in EXEC mode.

**show rpl as-path-set** [*name*] **states** | **brief**

## Syntax Description

<i>name</i>	(Optional) Name of the AS path set.
<b>states</b>	(Optional) Displays all unused, inactive, and active states.
<b>brief</b>	(Optional) Limits the display to a list of the names of all AS path sets without their configurations.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>summary</b> keyword was replaced with <b>brief</b> . The * keyword was removed and the <b>states</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the optional **brief** keyword to limit the display to a list of the names of all AS path sets without their configurations.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
RP/0/0/CPU0:router# show rpl route-policy policy_1
route-policy policy_1
```



```

if (destination in prefix_set_ex1) then
    set local-preference 100
endif
if (as-path in as_path_set_ex1) then
    set community (10:333) additive
endif
end-policy

```

Given this sample configuration, the **show rpl as-path-set as\_path\_set\_ex1** command displays the following information:

```

RP/0/0/CPU0:router# show rpl as-path-set as_path_set_ex1

as-path-set as_path_set_ex1
  ios-regex '^_65500_$',
  ios-regex '^_65501_$'
end-set

```

## Related Commands

Command	Description
<a href="#">show rpl community-set, on page 208</a>	Displays the configuration of a named community set.
<a href="#">show rpl extcommunity-set, on page 216</a>	Displays the configuration of a named extended community set.
<a href="#">show rpl route-policy, on page 256</a>	Displays the configuration of a named route policy.
<a href="#">show rpl prefix-set, on page 241</a>	Displays the configuration of a named prefix set.

# show rpl as-path-set attachpoints

To display all of the policies used at an attach point that reference the named AS path set, use the **show rpl as-path-set attachpoints** command in EXEC mode.

**show rpl as-path-set** *name* **attachpoints**

## Syntax Description

<i>name</i>	Name of an AS path set.
-------------	-------------------------

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl as-path-set attachpoints** command to display all policies used at an attach point that reference the named set either directly or indirectly.

The AS path set name is required.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
 !
 neighbor 10.0.101.3
```

```

remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!

```

```
RP/0/0/CPU0:router# show rpl route-policy policy_1
```

```

route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!

```

```
RP/0/0/CPU0:router# show rpl route-policy policy_2
```

```

route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!

```

Given this sample configuration, the **show rpl as-path-set as\_path\_set\_ex1 attachpoints** command displays the following information:

```
RP/0/0/CPU0:router# show rpl as-path-set as_path_set_ex1 attachpoints
```

```
BGP Attachpoint:Neighbor
```

Neighbor/Group	type	afi/safi	in/out	referring policy	attached policy
10.0.101.2	--	IPv4/uni	in	policy_1	policy_1
10.0.101.3	--	IPv4/uni	in	policy_2	policy_2

This table describes the significant fields shown in the display.

**Table 3: show rpl as-path-set attachpoints Field Descriptions**

Field	Description
BGP Attachpoint	Location of the attach point.
Neighbor/Group	IP address of the attach point on the neighbor.
type	Displays the address family mode.
afi/safi	Address family identifier or subsequent address family identifier.
in/out	Import or export policy.
referring policy	Policy that refers to the AS path set.
attached policy	Policy used at the attach point.

**Related Commands**

Command	Description
<a href="#">show rpl community-set attachpoints, on page 210</a>	Displays all the policies used at an attach point that reference the named community set.
<a href="#">show rpl route-policy attachpoints, on page 259</a>	Displays all the policies used at an attach point that reference the named policy.
<a href="#">show rpl prefix-set attachpoints, on page 243</a>	Displays all the policies used at an attach point that reference the named prefix set.

# show rpl as-path-set references

To list all of the policies that reference the named AS path set, use the **show rpl as-path-set references** command in EXEC mode.

**show rpl as-path-set** *name* **references** [**brief**]

## Syntax Description

<i>name</i>	Name of the prefix set.
<b>brief</b>	(Optional) Limits the output to just the brief table and not the detailed information for the named AS path set.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>summary</b> keyword was replaced with <b>brief</b> .

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl as-path-set references** command to display all policies that reference the named AS path set either directly or indirectly.

Use the optional **brief** keyword to limit the output to just a summary table and not the detailed information for the AS path set.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
```

```

!
neighbor 10.0.101.2
remote-as 100
address-family ipv4 unicast
route-policy policy_1 in
!
!
RP/0/0/CPU0:router# show rpl route-policy policy_1

route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy

```

Given this sample configuration, the **show rpl as-path-set as\_path\_set\_ex1 references** command displays the following information:

```

RP/0/0/CPU0:router# show rpl as-path-set as_path_set_ex1 references

Usage Direct -- Reference occurs in this policy
Usage Indirect -- Reference occurs via an apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached)
Status ACTIVE -- Policy is actively used at an attachpoint
Status INACTIVE -- Policy is applied by an unattached policy

      Usage/Status      count
-----
      Direct             1
      Indirect           0

      ACTIVE             1
      INACTIVE           0
      UNUSED             0

      route-policy      usage      policy status
-----
      policy_1          Direct     ACTIVE

```

This table describes the significant fields shown in the display.

**Table 4: show rpl as-path-set references Field Descriptions**

Field	Description
Usage/Status	Displays the usage and status of all policies that reference the AS path set.  Values for usage are Direct or Indirect.  Values for policy status are ACTIVE, INACTIVE, or UNUSED.
count	Number of policies that match each usage and status option.
route-policy	Name of the route policies that reference the AS path set.
usage	Type of usage for the policy.

Field	Description
policy status	Status of the policy.

**Related Commands**

Command	Description
<a href="#">show rpl community-set references, on page 213</a>	Lists all policies that reference the named community set.
<a href="#">show rpl route-policy references, on page 264</a>	Lists all policies that reference the named policy.
<a href="#">show rpl prefix-set references, on page 246</a>	Lists all policies that reference the named prefix set.

# show rpl community-set

To display the configuration of community sets, use the **show rpl community-set** command in EXEC mode.

**show rpl community-set** [*name*] **states** [**brief**]

## Syntax Description

<i>name</i>	(Optional) Name of the community set.
<b>states</b>	(Optional) Shows all unused, inactive, and active states.
<b>brief</b>	(Optional) Limits the display to a list of the names of all community sets without their configurations.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>summary</b> keyword was replaced with <b>brief</b> . The * keyword was removed and the <b>states</b> keyword was added.
Release 5.3.2	The command output was modified to display graceful maintenance feature information.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the optional **brief** keyword to limit the display to a list of the names of community sets without their configurations.

## Task ID

Task ID	Operations
route-policy	read



**Examples**

The following is the sample output of the show rpl community-set command with graceful maintenance feature attributes displayed:

```
RP/0/0/CPU0:R5#show rpl community-set
Thu Jan 29 17:55:04.792 PST
Listing for all Community Set objects

community-set gshut
  graceful-shutdown
end-set
```

**Examples**

This example shows the following sample configuration:

```
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
```

Given this sample configuration, the **show rpl community-set comm\_set\_ex2** command displays the following information:

```
RP/0/0/CPU0:router# show rpl community-set comm_set_ex2

community-set comm_set_ex2
  65501:1,
  65501:2,
  65501:3
end-set
```

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set, on page 200</a>	Displays the configuration of a named AS path set.
<a href="#">show rpl extcommunity-set, on page 216</a>	Displays the configuration of a named extended community set.
<a href="#">show rpl prefix-set, on page 241</a>	Displays the configuration of a named prefix set.
<a href="#">show rpl rd-set, on page 249</a>	Displays the configuration of a named RD set.
<a href="#">show rpl route-policy, on page 256</a>	Displays the configuration of a named route policy.

# show rpl community-set attachpoints

To display all the policies used at an attach point that reference the named community set, use the **show rpl community-set attachpoints** command in EXEC mode.

**show rpl community-set** *name* **attachpoints**

Syntax Description	
	<i>name</i> Name of a community set.

**Command Default** No default behavior or values

**Command Modes** EXEC

Command History	Release	Modification
	Release 3.2	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl community-set attachpoints** command to display all the policies used at an attach point that reference the named community set either directly or indirectly.

The community set name is required.

Task ID	Task ID	Operations
	route-policy	read

**Examples** This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
   route-policy policy_2 in
 !
 !
 !
```

```

route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then <<<<<
      set community (10:999) additive
    endif
  endif
end-policy
!

```

Given this sample configuration, the **show rpl community-set attachpoints** command displays the following information:

```

RP/0/0/CPU0:router# show rpl community-set ext_comm_set_rt_ex1 attachpoints
BGP Attachpoint:Neighbor
Neighbor/Group  type  afi/safi  in/out  referring policy  attached policy
-----
10.0.101.3      --   IPv4/uni  in      policy_2         policy_2

```

This table describes the significant fields shown in the display.

**Table 5: show rpl community-set attachpoints Field Descriptions**

Field	Description
BGP Attachpoint	Location of the attach point.
Neighbor/Group	IP address of the attach point on the neighbor.
type	Displays the address family mode.
afi/safi	Address family identifier or subsequent address family identifier.
in/out	Import or export policy.
referring policy	Policy that refers to the AS path set.
attached policy	Policy used at the attach point.

## Related Commands

Command	Description
<a href="#">show rpl as-path-set attachpoints, on page 202</a>	Displays all the policies used at an attach point that reference the named AS path set.
<a href="#">show rpl prefix-set attachpoints, on page 243</a>	Displays all the policies used at an attach point that reference the named prefix set.
<a href="#">show rpl rd-set attachpoints, on page 251</a>	Displays all the policies used at an attach point that reference the named RD set.

Command	Description
<a href="#">show rpl route-policy attachpoints, on page 259</a>	Displays all the policies used at an attach point that reference the named policy.

# show rpl community-set references

To list all the policies that reference the named community set, use the **show rpl community-set references** command in EXEC mode.

**show rpl community-set** *name* **references** [**brief**]

## Syntax Description

<i>name</i>	Name of a community set.
<b>brief</b>	(Optional) Limits the output to just the summary table and not the detailed information for the community set.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>summary</b> keyword was replaced with <b>brief</b> .

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl community-set references** command to display all the policies that reference the named community set.

Use the optional **brief** keyword to limit the output to just a summary table and not the detailed information for the community set.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!
route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy

```

Given this sample configuration, the **show rpl extcommunity-set comm\_set\_ex1 references** command displays the following information:

```
RP/0/0/CPU0:router# show rpl extcommunity-set comm_set_ex1 references
```

```

Usage Direct -- Reference occurs in this policy
Usage Indirect -- Reference occurs via an apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached)
Status ACTIVE -- Policy is actively used at an attachpoint
Status INACTIVE -- Policy is applied by an unattached policy

```

Usage/Status	count
-----	-----
Direct	1
Indirect	0
ACTIVE	1
INACTIVE	0
UNUSED	0

route-policy	usage	policy status
-----	-----	-----
policy_2	Direct	ACTIVE

This table describes the significant fields shown in the display.

**Table 6: show rpl community-set references Field Descriptions**

Field	Description
Usage/Status	Displays the usage and status of all policies that reference the community set.  Values for usage are Direct or Indirect.  Values for status are ACTIVE, INACTIVE, and UNUSED.
count	Number of policies that match each usage and status option.
route-policy	Name of the route policies that reference the community set.

Field	Description
usage	Type of usage for the policy.
policy status	Status of the policy.

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set references, on page 205</a>	Lists all policies that reference the named AS path set.
<a href="#">show rpl prefix-set references, on page 246</a>	Lists all policies that reference the named prefix set.
<a href="#">show rpl rd-set references, on page 253</a>	Lists all policies that reference the named RD set.
<a href="#">show rpl route-policy references, on page 264</a>	Lists all policies that reference the named policy.

# show rpl extcommunity-set

To display the configuration of extended community sets, use the **show rpl extcommunity-set** command in EXEC mode.

**show rpl extcommunity-set** [*name* [**attachpoints**| **references**]] [**cost**| **rt**| **soo**] [*name* ] [**brief**] [**states**]

## Syntax Description

<i>name</i>	(Optional) Name of the community set.
<b>attachpoints</b>	(Optional) Displays all attach points for this community set.
<b>references</b>	(Optional) Displays all policies that use this community set.
<b>cost</b>	(Optional) Displays all extended community cost sets.
<b>rt</b>	(Optional) Displays all extended community RT sets.
<b>soo</b>	(Optional) Displays all extended community SoO sets.
<b>brief</b>	(Optional) Limits the display to a list of the names of all extended community sets without their configurations.
<b>states</b>	(Optional) Displays all unused, inactive, and active states.

## Command Default

If an attachpoint or reference is not specified, all configured extended community sets are displayed  
If a cost, RT, or SoO sets is not specified, all configured extended community sets are displayed

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>summary</b> keyword was replaced with <b>brief</b> . The <b>*</b> keyword was removed and the <b>states</b> keyword was added.



Release	Modification
Release 3.3.0	<p>The following keywords were added:</p> <ul style="list-style-type: none"> <li>• <b>attachpoints</b></li> <li>• <b>references</b></li> <li>• <b>cost</b></li> <li>• <b>rt</b></li> <li>• <b>sso</b></li> </ul>

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the optional **brief** keyword to limit the display to a list of the names of extended community sets without their configurations.

### Task ID

Task ID	Operations
route-policy	read

### Examples

In the following example, the configuration of an extended community is displayed for the RT community set named `ext_comm_set_rt_ex1`:

```
RP/0/0/CPU0:router# show rpl extcommunity-set rt ext_comm_set_rt_ex1
ext_comm_set_rt_ex1
  1.2.3.4:34
end-set
!
```

In the following example, the configuration of an extended community is displayed with all RT set objects:

```
RP/0/0/CPU0:router# show rpl extcommunity-set rt
Listing for all Extended Community RT Set objects

extcommunity-set rt extrt1
  66:60001
end-set
!
extcommunity-set rt rtset1
  10:615,
  10:6150,
  15.15.15.15:15
end-set
!
extcommunity-set rt rtset3
  11:11,
  11.1.1.1:3
end-set
```

```

!
extcommunity-set rt extsoo1
  66:70001
end-set
!
extcommunity-set rt rtset11
  100:121,
  100:122,
  100:123,
  100:124,
  100:125,
  100:126,
  100:127,
  100:128,
  7.7.7.7:21
end-set
!

```

In the following example, the configuration of an extended community is displayed with all cost set objects:

```
RP/0/0/CPU0:router# show rpl extcommunity-set cost
```

Listing for all Extended Community COST Set objects

```

extcommunity-set cost costset1
  IGP:90:914,
  Pre-Bestpath:91:915
end-set
!
extcommunity-set cost costset2
  IGP:92:916,
  Pre-Bestpath:93:917,
  IGP:94:918,
  Pre-Bestpath:95:919
end-set
!

```

In the following example, the configuration of an extended community is displayed with all SoO set objects:

Extended Community SOO Set objects

```

extcommunity-set soo sooset1
  10:151,
  100.100.100.1:153
end-set
!
extcommunity-set soo sooset3
  11:11,
  11.1.1.1:3
end-set
!

```

## Related Commands

Command	Description
<a href="#">show rpl as-path-set, on page 200</a>	Displays the configuration of a named AS path set.
<a href="#">show rpl community-set, on page 208</a>	Displays the configuration of a named community set.
<a href="#">show rpl prefix-set, on page 241</a>	Displays the configuration of a named prefix set.
<a href="#">show rpl rd-set, on page 249</a>	Displays the configuration of a named RD set.
<a href="#">show rpl route-policy, on page 256</a>	Displays the configuration of a named route policy.



# show rpl inactive as-path-set

To display the AS path sets that are referenced by a policy but not in any policy that is used at an attach point, use the **show rpl inactive as-path-set** command in EXEC mode.

**show rpl inactive as-path-set [detail]**

Syntax Description	detail
	(Optional) Displays the content of the object and all referenced objects for inactive AS path sets.

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.2	This command was introduced.
	Release 3.3.0	Added the <b>detail</b> keyword.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Use the **show rpl inactive as-path-set** command to display all AS path sets that are not in use at an attach point either directly or indirectly but are referenced by at least one policy in the system.

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
 remote-as 12
 address-family ipv4 unicast
  route-policy policy_2 in
!
!
route-policy sample
  if (destination in sample) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_5
  apply sample1
  apply policy_3
end-policy

```

Given this sample configuration, the **show rpl inactive as-path-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive as-path-set
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```
The following as-path-sets are INACTIVE
-----
```

as\_path\_set\_ex2

**Related Commands**

Command	Description
<a href="#">show rpl inactive community-set, on page 223</a>	Displays the community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive extcommunity-set, on page 226</a>	Displays the extended community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive prefix-set, on page 229</a>	Displays the prefix sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive rd-set, on page 232</a>	Displays the RD sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive route-policy, on page 234</a>	Displays the route policies that are referenced by a policy but not in any policy that is used at an attach point.

## show rpl inactive community-set

To display the community sets that are referenced by a policy but not any policy that is used at an attach point, use the **show rpl inactive community-set** command in EXEC mode.

**show rpl inactive community-set [detail]**

### Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for inactive community sets.
---------------	---

### Command Default

No default behavior or values

### Command Modes

EXEC

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl inactive community-set** command to display all community sets that are not in use at an attach point either directly or indirectly but are referenced by at least one policy in the system.

### Task ID

Task ID	Operations
route-policy	read

### Examples

This example shows the following sample configuration:

```
router bgp 2
  address-family ipv4 unicast
  !
  neighbor 10.0.101.2
    remote-as 100
    address-family ipv4 unicast
    route-policy policy_1 in
  !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
route-policy sample2
  if (destination in sample2) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_5
  apply sample2
  apply policy_3
end-policy

```

Given this sample configuration, the **show rpl inactive community-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive community-set
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```
The following community-sets are INACTIVE
```

```
-----
```



comm\_set\_ex2

**Related Commands**

Command	Description
<a href="#">show rpl inactive as-path-set, on page 220</a>	Displays the AS path sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive extcommunity-set, on page 226</a>	Displays the extended community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive prefix-set, on page 229</a>	Displays the prefix sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive rd-set, on page 232</a>	Displays the RD sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive route-policy, on page 234</a>	Displays the route policies that are referenced by a policy but not in any policy that is used at an attach point.

# show rpl inactive extcommunity-set

To display the extended community sets that are referenced by a policy but not in any policy that is used at an attach point, use the **show rpl inactive extcommunity-set** command in EXEC mode.

**show rpl inactive extcommunity-set [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for inactive extended community sets.
---------------	--

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl inactive extcommunity-set** command to display all extended community sets that are not in use at an attach point either directly or indirectly but are referenced by at least one policy in the system.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
 remote-as 100
 address-family ipv4 unicast
 route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
  route-policy policy_2 in
!
!
route-policy sample3
  if (destination in sample3) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_5
  apply sample3
  apply policy_3
end-policy

```

Given this sample configuration, the **show rpl inactive extcommunity-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive extcommunity-set
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```
The following extcommunity-sets are INACTIVE
```

```
ext_comm_set_rt_ex2
```

**Related Commands**

Command	Description
<a href="#">show rpl inactive as-path-set, on page 220</a>	Displays the AS path sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive community-set, on page 223</a>	Displays the community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive prefix-set, on page 229</a>	Displays the prefix sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive rd-set, on page 232</a>	Displays the RD sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive route-policy, on page 234</a>	Displays the route policies that are referenced by a policy but not in any policy that is used at an attach point.

# show rpl inactive prefix-set

To display the prefix sets that are referenced by a policy but not in any policy that is used at an attach point, use the **show rpl inactive prefix-set** command in EXEC mode.

## show rpl inactive prefix-set [detail]

<b>Syntax Description</b>	<b>detail</b> (Optional) Displays the content of the object and all referenced objects for inactive prefix sets.
---------------------------	--

<b>Command Default</b>	No default behavior or values
------------------------	-------------------------------

<b>Command Modes</b>	EXEC
----------------------	------

<b>Command History</b>	Release	Modification
	Release 3.2	This command was introduced.
	Release 3.3.0	The <b>detail</b> keyword was added.

<b>Usage Guidelines</b>	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
-------------------------	---

Use the **show rpl inactive prefix-set** command to display all prefix sets that are not in use at an attach point either directly or indirectly but are referenced by at least one policy in the system.

<b>Task ID</b>	Task ID	Operations
	route-policy	read

<b>Examples</b>	This example shows the following sample configuration:
-----------------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
route-policy sample4
  if (destination in sample4) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if destination in prefix_set_ex1 then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_5
  apply sample4
  apply policy_3
end-policy

```

Given this sample configuration, the **show rpl inactive prefix-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive prefix-set
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```
The following prefix-sets are INACTIVE
```

```
-----
sample4
```

```
prefix_set_ex2
```

**Related Commands**

Command	Description
<a href="#">show rpl inactive as-path-set, on page 220</a>	Displays the AS path sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive community-set, on page 223</a>	Displays the community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive extcommunity-set, on page 226</a>	Displays the extended community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive route-policy, on page 234</a>	Displays the route policies that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive rd-set, on page 232</a>	Displays the RD sets that are referenced by a policy but not in any policy that is used at an attach point.

# show rpl inactive rd-set

To display the route distinguisher (RD) sets that are referenced by a policy but not in any policy that is used at an attach point, use the **show rpl inactive rd-set** command in EXEC mode.

**show rpl inactive rd-set [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for inactive RD sets.
---------------	--

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl inactive rd-set** command to display all RD sets that are not in use at an attach point either directly or indirectly but are referenced by at least one policy in the system.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
rd-set rdset1
  10:151,
  100.100.100.1:153,
  100.100.100.62/31:63
end-set
!
rd-set rdset2
  10:152,
  100.100.100.1:154,
  100.100.100.62/31:89
```



```
end-set
!
```

Given this sample configuration, the **show rpl inactive rd-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive rd-set
```

```
ACTIVE -- Referenced by at least one policy which is attached INACTIVE -- Only referenced
by policies which are not attached UNUSED -- Not attached (directly or indirectly) and not
referenced
```

```
The following rd-sets are INACTIVE
```

```
-----
rdset1
rdset2
```

## Related Commands

Command	Description
<a href="#">show rpl inactive as-path-set, on page 220</a>	Displays the AS path sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive community-set, on page 223</a>	Displays the community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive extcommunity-set, on page 226</a>	Displays the extended community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive prefix-set, on page 229</a>	Displays the prefix sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive route-policy, on page 234</a>	Displays the route policies that are referenced by a policy but not in any policy that is used at an attach point.

# show rpl inactive route-policy

To display the route policies that are referenced by a policy but not in any policy that is used at an attach point, use the **show rpl inactive route-policy** command in EXEC mode.

**show rpl inactive route-policy [detail]**

Syntax Description	detail	(Optional) Displays the content of the object and all referenced objects for inactive route policies.
--------------------	--------	---

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.2	This command was introduced. The <b>policy</b> keyword was changed to <b>route-policy</b> .
	Release 3.3.0	The <b>detail</b> keyword was added.

Usage Guidelines	<p>To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.</p> <p>Use the <b>show rpl inactive route-policy</b> command to display all policies that are not in use at an attach point either directly or indirectly but are referenced by at least one other policy in the system.</p>
------------------	---

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
 remote-as 100
 address-family ipv4 unicast
```

```

    route-policy policy_1 in
    !
    !
neighbor 10.0.101.3
    remote-as 12
    address-family ipv4 unicast
    route-policy policy_2 in
    !
    !
route-policy sample3
    if (destination in sample3) then
        drop
    endif
end-policy
!
route-policy policy_1
    if (destination in prefix_set_ex1) then
        set local-preference 100
    endif
    if (as-path in as_path_set_ex1) then
        set community (10:333) additive
    endif
end-policy
!
route-policy policy_2
    if destination in prefix_set_ex1 then
        if (community matches-any comm_set_ex1) then
            set community (10:666) additive
        endif
        if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
            set community (10:999) additive
        endif
    endif
end-policy
!
route-policy policy_3
    if (destination in prefix_set_ex2) then
        set local-preference 100
    endif
    if (as-path in as_path_set_ex2) then
        set community (10:333) additive
    endif
end-policy
!
route-policy policy_4
    if (destination in prefix_set_ex2) then
        if (community matches-any comm_set_ex2) then
            set community (10:666) additive
        endif
        if (extcommunity matches-any ext_comm_set_rt_ex2) then
            set community (10:999) additive
        endif
    endif
end-policy
!
route-policy policy_5
    apply sample3
    apply policy_3
end-policy

```

Given this sample configuration, the **show rpl inactive route-policy** command displays the following information:

```
RP/0/0/CPU0:router# show rpl inactive route-policy
```

```
ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced
```

The following policies are (INACTIVE)

```
-----
sample3
policy_3
```

**Related Commands**

Command	Description
<a href="#">show rpl inactive as-path-set, on page 220</a>	Displays the AS path sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive community-set, on page 223</a>	Displays the community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive extcommunity-set, on page 226</a>	Displays the extended community sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive prefix-set, on page 229</a>	Displays the prefix sets that are referenced by a policy but not in any policy that is used at an attach point.
<a href="#">show rpl inactive rd-set, on page 232</a>	Displays the RD sets that are referenced by a policy but not in any policy that is used at an attach point.

# show rpl maximum

To display the maximum limits for lines of configuration and number of policies, use the **show rpl maximum** command in EXEC mode.

**show rpl maximum** [**lines**|**policies**]

## Syntax Description

<b>lines</b>	(Optional) Displays the number of lines of configuration limit.
<b>policies</b>	(Optional) Displays the number of policies limit.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl maximum** command to display the current total, current limit, and maximum limit for lines of configuration and policies.

Use the optional **lines** keyword to limit the display to the number of lines of configuration limits. Use the optional **policies** keyword to limit the display to the number of policies limits.

## Task ID

Task ID	Operations
route-policy	read

## Examples

The following example shows sample output from the **show rpl maximum** command:

```
RP/0/0/CPU0:router# show rpl maximum
                        Current      Current      Max
                        Total        Limit        Limit
-----
Lines of configuration      3      65536      131072
Policies                   1       3500       5000
```

```
Compiled policies size (kB)          0
```

[Table 7: show rpl maximum Field Descriptions, on page 238](#) describes the significant fields shown in the display.

**Table 7: show rpl maximum Field Descriptions**

Field	Description
Lines of configuration	Displays the current total, current limit, and maximum limit of lines for the policy.
Policies	Displays the current total, current limit, and maximum limit of policies.
Compiled policies size (kB)	Displays the current compiled total for policies in kilobytes.

#### Related Commands

Command	Description
<a href="#">rpl maximum, on page 126</a>	Configures the maximum number of lines of configuration and number of policies.

# show rpl policy-global references

To display policy-global definitions, use the **show rpl policy-global references** command in EXEC mode.

**show rpl policy-global references [brief]**

Syntax Description	<b>brief</b> (Optional) Limits the display to a list of the policy names.
--------------------	---

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.6.0	This command was introduced.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
policy-global
  infinity '16'
end-global
!
route-policy set-rip-unreachable
  set rip-metric $infinity
end-policy
!
```

Given this sample configuration, the **show rpl policy-global references** command displays the following information:

```
RP/0/0/CPU0:router# show rpl policy-global references
```

```
Usage Direct -- Reference occurs in this policy Usage Indirect -- Reference occurs via an
```

apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached) Status ACTIVE -- Policy is actively used at an attachpoint Status INACTIVE -- Policy is applied by an unattached policy

Usage/Status		count
-----		
Direct		1
Indirect		0
ACTIVE		0
INACTIVE		0
UNUSED		1
Usage	Status	Route-policy
-----		
Direct	UNUSED	set-rip-unreachable



# show rpl prefix-set

To display the configuration of prefix sets, use the **show rpl prefix-set** command in EXEC mode.

**show rpl prefix-set** [*name*] **states** **brief**

## Syntax Description

<i>name</i>	(Optional) Name of the prefix set.
<b>states</b>	(Optional) Shows all unused, inactive, and active states.
<b>brief</b>	(Optional) Limits the display to a list of the names of all extended community sets without their configurations.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>summary</b> keyword was replaced with <b>brief</b> .
	The * keyword was removed and the <b>states</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Because sets cannot hierarchically reference other sets or policies, no **detail** keyword exists as with the **show rpl policy** command.

## Task ID

Task ID	Operations
route-policy	read

## Examples

In the following example, the configuration of prefix set pset1 is displayed:

```
RP/0/0/CPU0:router# show rpl prefix-set pset1
!
```

**show rpl prefix-set**

```

prefix-set pset1
 10.0.0.1/0,
 10.0.0.2/0 ge 25 le 32,
 10.0.0.5/8 ge 8 le 32,
 10.168.0.0/16 ge 16 le 32,
 172.16.0.9/20 ge 20 le 32,
 192.168.0.5/20 ge 20 le 32
end-set

```

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set, on page 200</a>	Displays the configuration of a named AS path set.
<a href="#">show rpl community-set, on page 208</a>	Displays the configuration of a named community set.
<a href="#">show rpl extcommunity-set, on page 216</a>	Displays the configuration of a named extended community set.
<a href="#">show rpl route-policy, on page 256</a>	Displays the configuration of a named route policy.

# show rpl prefix-set attachpoints

To display all the policies used at an attach point that reference the named prefix set, use the **show rpl prefix-set attachpoints** command in EXEC mode.

**show rpl prefix-set** *name* **attachpoints**

Syntax Description	<i>name</i> Name of a prefix set.
--------------------	--------------------------------------

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.2	This command was introduced.

Usage Guidelines	To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
------------------	---

Use the **show rpl prefix-set attachpoints** command to display all the policies used at an attach point that reference the named prefix set either directly or indirectly.

The prefix set name is required.

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
 !
 neighbor 10.0.101.3
```

```

remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy

```

Given this sample configuration, the **show rpl prefix-set prefix\_set\_ex1 attachpoints** command displays the following information:

```
RP/0/0/CPU0:router# show rpl prefix-set prefix_set_ex1 attachpoints
```

```
BGP Attachpoint:Neighbor
```

Neighbor/Group	type	afi/safi	in/out	referring policy	attached policy
10.0.101.2	--	IPv4/uni	in	policy_1	policy_1
10.0.101.3	--	IPv4/uni	in	policy_2	policy_2

This table describes the significant fields shown in the display.

**Table 8: show rpl prefix-set attachpoints Field Descriptions**

Field	Description
BGP Attachpoint	Location of the attach point.
Neighbor/Group	IP address of the attach point on the neighbor.
type	Address family mode.
afi/safi	Address family identifier or subsequent address family identifier.
in/out	Import or export policy.
referring policy	Policy that refers to the AS path set.
attached policy	Policy used at the attach point.

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set attachpoints, on page 202</a>	Displays all the policies used at an attach point that reference the named AS path set.
<a href="#">show rpl community-set attachpoints, on page 210</a>	Displays all the policies used at an attach point that reference the named community set.
<a href="#">show rpl route-policy attachpoints, on page 259</a>	Displays all the policies used at an attach point that reference the named policy.

# show rpl prefix-set references

To list all the policies that reference the named prefix set, use the **show rpl prefix-set references** command in EXEC mode.

**show rpl prefix-set** *name* **references** [**brief**]

## Syntax Description

<i>name</i>	Name of the prefix set.
<b>brief</b>	(Optional) Limits the output to just a summary table and not the detailed information for the named prefix set.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>summary</b> keyword was replaced with <b>brief</b> .

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl prefix-set references** command to list all the policies that reference the named prefix set.

Use the optional **brief** keyword to limit the output to just a summary table and not the detailed information for the named prefix set.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
prefix-set ten-net
 10.0.0.0/16 le 32
end-set
prefix-set too-specific
```

```

0.0.0.0/0 ge 25 le 32
end-set
route-policy example-one
  if destination in ten-net then
    drop
  else
    set local-preference 200
    apply set-comms
  endif
end-policy
route-policy set-comms
  set community (10:1234) additive
end-policy

route-policy example-three
  if destination in too-specific then
    drop
  else
    apply example-one
    pass
  endif
end-policy

```

The following example displays information showing the usage and status of each policy that references the prefix set ten-net. The **brief** keyword limits the display to just a summary table and not the detailed information for the prefix set.

```
RP/0/0/CPU0:router# show rpl prefix-set ten-net references brief
```

```

Usage Direct -- Reference occurs in this policy
Usage Indirect -- Reference occurs via an apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached)
Status ACTIVE -- Policy is actively used at an attachpoint
Status INACTIVE -- Policy is applied by an unattached policy

```

Usage/Status	count
Direct	1
Indirect	1
ACTIVE	0
INACTIVE	1
UNUSED	1

This table describes the significant fields shown in the display.

**Table 9: show rpl prefix-set name references Field Descriptions**

Field	Description
Usage/Status	Displays the usage and status of all policies that reference the prefix set.
count	Number of policies that match each usage and status option.

## Related Commands

Command	Description
<a href="#">show rpl as-path-set references</a> , on page 205	Lists all the policies that reference the named AS path set.

Command	Description
<a href="#">show rpl community-set references, on page 213</a>	Lists all the policies that reference the named community set.
<a href="#">show rpl route-policy references, on page 264</a>	Lists all the policies that reference the named policy.



# show rpl rd-set

To display the configuration of route distinguisher (RD) sets, use the **show rpl rd-set** command in EXEC mode.

**show rpl rd-set** [*name*] **states** [**brief**]

## Syntax Description

<i>name</i>	(Optional) Name of the RD set.
<b>states</b>	(Optional) Shows all unused, inactive, and active states.
<b>brief</b>	(Optional) Limits the display to a list of the names of all RD sets without their configurations.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Because sets cannot hierarchically reference other sets or policies, no **detail** keyword exists as with the **show rpl policy** command.

## Task ID


Task ID	Operations
route-policy	read

## Examples

In the following example, the configuration of RD set rdset1 is displayed:

```
RP/0/0/CPU0:router# show rpl rd-set rdset1

rd-set rdset1
  10:151,
  100.100.100.1:153,
```

 show rpl rd-set

```
100.100.100.62/31:63
end-set
```

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set, on page 200</a>	Displays the configuration of a named AS path set.
<a href="#">show rpl community-set, on page 208</a>	Displays the configuration of a named community set.
<a href="#">show rpl extcommunity-set, on page 216</a>	Displays the configuration of a named extended community set.
<a href="#">show rpl prefix-set, on page 241</a>	Displays the configuration of a named prefix set.
<a href="#">show rpl route-policy, on page 256</a>	Displays the configuration of a named route policy.

# show rpl rd-set attachpoints

To display all the policies used at an attach point that reference the named route distinguisher (RD) set, use the **show rpl rd-set attachpoints** command in EXEC mode.

**show rpl rd-set** *name* **attachpoints**

## Syntax Description

<i>name</i>	Name of an RD set.
-------------	--------------------

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl rd-set attachpoints** command to display all the policies used at an attach point that reference the named RD set either directly or indirectly.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
route-policy rdsetmatch
  if rd in rdset1 then
    set community (10:112)
  elseif rd in rdset2 then
    set community (10:223)
  endif
end-policy

router bgp 10
address-family vpnv4 unicast
exit
neighbor 10.0.101.1
```

**show rpl rd-set attachpoints**

```

remote-as 11
address-family vpnv4 unicast
route-policy rdsetmatch in
!

```

Given this sample configuration, the **show rpl rd-set rdset1 attachpoints** command displays the following information:

```
RP/0/0/CPU0:router# show rpl rd-set rdset attachpoints
```

```
BGP Attachpoint: Neighbor
```

```

Neighbor/Group  type  afi/safi  in/out  vrf name
-----
10.0.101.1      --   IPv4/vpn  in      default

```

This table describes the significant fields shown in the display.

**Table 10: show rpl rd-set attachpoints Field Descriptions**

Field	Description
Neighbor/Group	BGP neighbor or neighbor group where the specified RD is used.
afi/safi	BGP address family or subaddress family where the RD set is used.
in/out	Direction
vrf name	VRF name where the RD set is used.

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set attachpoints, on page 202</a>	Displays all the policies used at an attach point that reference the named AS path set.
<a href="#">show rpl community-set attachpoints, on page 210</a>	Displays all the policies used at an attach point that reference the named community set.
<a href="#">show rpl prefix-set attachpoints, on page 243</a>	Displays all the policies used at an attach point that reference the named prefix set.
<a href="#">show rpl route-policy attachpoints, on page 259</a>	Displays all the policies used at an attach point that reference the named policy.

# show rpl rd-set references

To list all the policies that reference the named route distinguisher (RD) set, use the **show rpl rd-set references** command in EXEC mode.

**show rpl rd-set** *name* **references** [**brief**]

## Syntax Description

<i>name</i>	Name of the RD set.
<b>brief</b>	(Optional) Limits the output to just a summary table and not the detailed information for the RD set.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl rd-set references** command to list all the policies that reference the named RD set.

Use the optional **brief** keyword to limit the output to just a summary table and not the detailed information for the named RD set.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
route-policy rdsetmatch
  if rd in rdset1 then
    set community (10:112)
  elseif rd in rdset2 then
    set community (10:223)
  endif
```

```

end-policy
!
router bgp 10
  address-family vpnv4 unicast
  !
  neighbor 10.0.101.1
    remote-as 11
    address-family vpnv4 unicast
    route-policy rdsetmatch in
  !

```

Given this sample configuration, the **show rpl rd-set rdset1 references** command displays the following information:

```
RP/0/0/CPU0:router# show rpl rd-set rdset1 references
```

```

Usage Direct -- Reference occurs in this policy
Usage Indirect -- Reference occurs via an apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached)
Status ACTIVE -- Policy is actively used at an attachpoint
Status INACTIVE -- Policy is applied by an unattached policy

```

Usage/Status	count
-----	-----
Direct	1
Indirect	0
ACTIVE	1
INACTIVE	0
UNUSED	0

route-policy	usage	policy status
-----	-----	-----
rdsetmatch	Direct	ACTIVE

This table describes the significant fields shown in the display.

**Table 11: show rpl rd-set name references Field Descriptions**

Field	Description
route-policy	Name of the route policy.
usage	Type of reference usage for the route policy.
policy status	Status of the route policy.

## Related Commands

Command	Description
<a href="#">show rpl as-path-set references, on page 205</a>	Lists all the policies that reference the named AS path set.
<a href="#">show rpl community-set references, on page 213</a>	Lists all the policies that reference the named community set.

Command	Description
<a href="#">show rpl prefix-set references, on page 246</a>	Lists all the policies that reference the named prefix set.
<a href="#">show rpl route-policy references, on page 264</a>	Lists all policies that reference the named policy.

# show rpl route-policy

To display the configuration of route policies, use the **show rpl route-policy** command in EXEC mode.

**show rpl route-policy** [*name* [**detail**]| **states**| **brief**]

## Syntax Description

<i>name</i>	(Optional) Name of a route policy.
<b>detail</b>	(Optional) Displays the configuration of all policies and sets that a policy uses.
<b>states</b>	(Optional) Shows all unused, inactive, and active states.
<b>brief</b>	(Optional) Limits the display to a list of the names of all extended community sets without their configurations.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>summary</b> keyword was replaced with <b>brief</b> . The <b>policy</b> keyword was changed to <b>route-policy</b> . The <b>*</b> keyword was removed and the <b>states</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the optional **brief** keyword to limit the display to a list of the names of policies without their configurations.

## Task ID

Task ID	Operations
route-policy	read



**Examples**

In the following example, the configuration of a route policy named policy\_1 is displayed.

```
RP/0/0/CPU0:router# show rpl route-policy policy_1

route-policy policy_1
  if destination in prefix_set_1 and not destination in sample1 then
    if as-path in aspath_set_1 then
      set local-preference 300
      set origin igp
    elseif as-path in as_allowed then
      set local-preference 400
      set origin igp
    else
      set origin igp
    endif
  else
    drop
  endif
set med 120
set community (8660:612) additive
apply set_lpref_from_comm
end-policy
```

If the optional **detail** keyword is used, all routing policy language (RPL) policies and sets that route policy policy\_1 uses are displayed, as shown in the following example.

```
RP/0/0/CPU0:router# show rpl route-policy policy_1 detail

!
prefix-set sample1
  0.0.0.0/0,
  0.0.0.0/0 ge 25 le 32,
  10.0.0.0/8 ge 8 le 32,
  192.168.0.0/16 ge 16 le 32,
  224.0.0.0/20 ge 20 le 32,
  240.0.0.0/20 ge 20 le 32
end-set
!
prefix-set prefix_set_1
  10.0.0.1/24 ge 24 le 32,
  10.0.0.5/24 ge 24 le 32,
  172.16.0.1/24 ge 24 le 32,
  172.16.5.5/24 ge 24 le 32,
  172.16.20.10/24 ge 24 le 32,
  172.30.0.1/24 ge 24 le 32,
  10.0.20.10/24 ge 24 le 32,
  172.18.0.5/24 ge 24 le 32,
  192.168.0.1/24 ge 24 le 32,
  192.168.20.10/24 ge 24 le 32,
  192.168.200.10/24 ge 24 le 32,
  192.168.255.254/24 ge 24 le 32
end-set
!
as-path-set as_allowed
  ios-regex '.*_1239_.*',
  ios-regex '.*_3561_.*',
  ios-regex '.*_701_.*',
  ios-regex '.*_666_.*',
  ios-regex '.*_1755_.*',
  ios-regex '.*_1756_.*'
end-set
!
as-path-set aspath_set_1
  ios-regex '9148_',
  ios-regex '5870_',
  ios-regex '2408_',
  ios-regex '2531_',
  ios-regex '197_'
```

```

    ios-regex '_2992_'
end-set
!
route-policy set_lpref_from_comm
  if community matches-any (2:50) then
    set local-preference 50
  elseif community matches-any (2:60) then
    set local-preference 60
  elseif community matches-any (2:70) then
    set local-preference 70
  elseif community matches-any (2:80) then
    set local-preference 80
  elseif community matches-any (2:90) then
    set local-preference 90
  endif
end-policy
!
route-policy policy_1
  if destination in prefix_set_1 and not destination in sample1 then
    if as-path in aspath_set_1 then
      set local-preference 300
      set origin igp
    elseif as-path in as_allowed then
      set local-preference 400
      set origin igp
    else
      set origin igp
    endif
  else
    drop
  endif
  set med 120
  set community (8660:612) additive
  apply set_lpref_from_comm
end-policy

```

## Related Commands

Command	Description
<a href="#">show rpl as-path-set, on page 200</a>	Displays the configuration of a named AS path set.
<a href="#">show rpl community-set, on page 208</a>	Displays the configuration of a named community set.
<a href="#">show rpl extcommunity-set, on page 216</a>	Displays the configuration of a named extended community set.
<a href="#">show rpl prefix-set, on page 241</a>	Displays the configuration of a named prefix set.

# show rpl route-policy attachpoints

To display all the policies used at an attach point that reference the named policy, use the **show rpl route-policy attachpoints** command in EXEC mode.

**show rpl route-policy** *name* **attachpoints**

Syntax Description	<i>name</i>	Name of a policy.
--------------------	-------------	-------------------

Command Default	No default behavior or values
-----------------	-------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 3.2	This command was introduced. The <b>policy</b> keyword was changed to <b>route-policy</b> .

Usage Guidelines	<p>To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.</p>
------------------	--

Use the **show rpl route-policy attachpoints** command to display all the policies used at an attach point that reference the named policy either directly or indirectly.

The policy name is required.

Task ID	Task ID	Operations
	route-policy	read

Examples	This example shows the following sample configuration:
----------	--

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!

RP/0/0/CPU0:router# show rpl route-policy policy_1

route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
RP/0/0/CPU0:router# show rpl route-policy policy_2

route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!

```

The following command displays the route policy attach points for policy\_2:

```

RP/0/0/CPU0:router# show rpl route-policy policy_2 attachpoints

BGP Attachpoint: Neighbor

Neighbor/Group  type  afi/safi  in/out  vrf name
-----
10.0.101.2      --    IPv4/uni  in      default
10.0.101.2      --    IPv4/uni  out     default

```

This table describes the significant fields shown in the display.

**Table 12: show rpl route-policy attachpoints Field Descriptions**

Field	Description
BGP Attachpoint	Location of the attach point.
Neighbor/Group	IP address of the attach point on the neighbor.
type	Displays the address family mode.
afi/safi	Address family identifier or subsequent address family identifier.
vrf name	Name of the VPN routing and forwarding (VRF) instance.

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set attachpoints, on page 202</a>	Displays all the policies used at an attach point that reference the named AS path set.
<a href="#">show rpl community-set attachpoints, on page 210</a>	Displays all the policies used at an attach point that reference the named community set.
<a href="#">show rpl prefix-set attachpoints, on page 243</a>	Displays all the policies used at an attach point that reference the named prefix set.

# show rpl route-policy inline

To display all policies and sets that a policy uses expanded inline, use the **show rpl route-policy inline** command in EXEC mode.

**show rpl route-policy** *name* **inline**

Syntax Description	
	<i>name</i> Name of a policy.

<b>Command Default</b>	No default behavior or values
------------------------	-------------------------------

<b>Command Modes</b>	EXEC
----------------------	------

Command History	Release	Modification
	Release 3.5.0	This command was introduced.

**Usage Guidelines**

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl route-policy inline** command to examine the configuration of a specified route policy. All policies and sets that a policy uses are gathered together and displayed expanded inline.

The policy name is required.

Task ID	Task ID	Operations
	route-policy	read

## Examples

The following command displays the route policy policy\_1:

```
RP/0/0/CPU0:router# show rpl route-policy policy_1
!
route-policy policy_1
  if destination in prefix_set_1 and not destination in martians then
    if as-path in aspath_set_1 then
      set local-preference 300
      set origin igp
    elseif as-path in as_allowed then
      set local-preference 400
```

```

        set origin igp
    else
        set origin igp
    endif
else
    drop
endif
set med 120
set community (8660:612) additive
apply set_lpref_from_comm
end-policy

```

The following command displays the route policy policy\_1 and all the other sets or policies it refers too inline. Adding the inline keyword causes the configuration to be displayed inline for all RPL objects that the route-policy policy\_1 uses.

RP/0/0/CPU0:router#show rpl policy policy\_1 inline

```

route-policy policy_1
  if destination in (91.5.152.0/24 ge 24 le 32, 91.220.152.0/24 ge 24 le 32, 61.106.52.0/24
ge 24 le 32, 222.168.199.0/24
ge 24 le 32, 93.76.114.0/24 ge 24 le 32, 41.195.116.0/24 ge 24 le 32, 35.92.152.0/24 ge
24 le 32, 143.144.96.0/24 ge 24
le 32, 79.218.81.0/24 ge 24 le 32, 75.213.219.0/24 ge 24 le 32, 178.220.61.0/24 ge 24 le
32, 27.195.65.0/24 ge 24 le 32)
  and not destination in (0.0.0.0/0, 0.0.0.0/0 ge 25 le 32, 10.0.0.0/8 ge 8 le 32,
192.168.0.0/16 ge 16 le 32, 224.0.0.0/20
ge 20 le 32, 240.0.0.0/20 ge 20 le 32) then
    if as-path in (ios-regex '_9148_', ios-regex '_5870_', ios-regex '_2408_', ios-regex
' _2531_', ios-regex ' _197_',
ios-regex ' _2992_') then
      set local-preference 300
      set origin igp
    elseif as-path in
(ios-regex '.*_1239_.*', ios-regex '.*_3561_.*', ios-regex '.*_701_.*', ios-regex
'.*_666_.*', ios-regex '.*_1755_.*',
ios-regex '.*_1756_.*') then
      set local-preference 400
      set origin igp
    else
      set origin igp
    endif
  else
    drop
  endif
set med 120
set community (8660:612) additive
# apply set_lpref_from_comm
if community matches-any (2:50) then
  set local-preference 50
elseif community matches-any (2:60) then
  set local-preference 60
elseif community matches-any (2:70) then
  set local-preference 70
elseif community matches-any (2:80) then
  set local-preference 80
elseif community matches-any (2:90) then
  set local-preference 90
endif
# end-apply set_lpref_from_comm
end-policy

```

# show rpl route-policy references

To list all the policies that reference the named policy, use the **show rpl route-policy references** command in EXEC mode.

**show rpl route-policy** *name* **references** [**brief**]

## Syntax Description

<i>name</i>	Name of a prefix set.
<b>brief</b>	(Optional) Limits the output to just a summary table and not the detailed information for the named policy.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>summary</b> keyword was replaced with <b>brief</b> .
	The <b>policy</b> keyword was replaced with <b>route-policy</b> .

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl route-policy references** command to list all the policies that reference the named policy.

Use the optional **brief** keyword to limit the output to just a summary table and not the detailed information for the policy.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
prefix-set ten-net
 10.0.0.0/16 le 32
```



```

end-set
prefix-set too-specific
  0.0.0.0/0 ge 25 le 32
end-set
route-policy example-one
  if destination in ten-net then
    drop
  else
    set local-preference 200
    apply set-comms
  endif
end-policy
route-policy set-comms
  set community (10:1234) additive
end-policy
route-policy example-three
  if destination in too-specific then
    drop
  else
    apply example-one
    pass
  endif
end-policy

```

The following command displays information about the policy set-comms and how it is referenced:

```
RP/0/0/CPU0:router# show rpl route-policy set-comms references
```

Usage Direct -- Reference occurs in this policy

Usage Indirect -- Reference occurs via an apply statement

Status UNUSED -- Policy is not in use at an attachpoint (unattached)

Status ACTIVE -- Policy is actively used at an attachpoint

Status INACTIVE -- Policy is applied by an unattached policy

Usage/Status	count
Direct	1
Indirect	1
ACTIVE	0
INACTIVE	1
UNUSED	1

route-policy	usage	policy status
example-one	Direct	INACTIVE
example-three	Indirect	UNUSED

The direct usage indicates that the route policy example-one directly applies the policy set-comms, that is, example-one has a line in the form `apply set-comms`. The usage Indirect indicates that the route policy example-three does not directly apply the route policy set-comms. However, the route policy example-three does apply the policy example-one, which in turn applies the policy set-comms, so there is an indirect reference from example-three to the route policy set-comms.

The status column indicates one of three states. A policy is active if it is in use at an attach point. In the example provided, neither example-one nor example-three is in use at an attach point, which leaves two possible states: UNUSED or INACTIVE. The route policy example-one is inactive because it has some other policy (example-three) that references it, but neither example-one nor any of the policies that reference it (example-one) are in use at an attach point. The route policy example-three has a status of unused because it is not used at an attach point and no other route policies in the system refer to it.

This table describes the significant fields shown in the display.

**Table 13: show rpl route-policy references Field Descriptions**

Field	Description
Usage/Status	Displays the usage and status of all policies that reference the specified policy.  Values for usage are Direct or Indirect.  Values for status are ACTIVE, INACTIVE, and UNUSED.
count	Number of policies that match each usage and status option.
route-policy	One name for multiple policies that reference the specified policy.
usage	Type of usage for the policy.
policy status	Status of the policy.

**Related Commands**

Command	Description
<a href="#">show rpl as-path-set references, on page 205</a>	Lists all policies that reference the named AS path set.
<a href="#">show rpl community-set references, on page 213</a>	Lists all policies that reference the named community set.
<a href="#">show rpl prefix-set references, on page 246</a>	Lists all policies that reference the named prefix set.

# show rpl route-policy uses

To display information about a specified named policy, use the **show rpl route-policy uses** command in EXEC mode.

**show rpl route-policy** *name* **uses** {**policies**|**sets**|**all**} [**direct**]

## Syntax Description

<i>name</i>	Name of a policy.
<b>policies</b>	Generates a list of all policies that the named policy uses.
<b>sets</b>	Lists all named sets that are used by the policy.
<b>all</b>	Generates a list of both sets and policies that the named policy references.
<i>direct</i>	(Optional) Lists only the policies or sets used directly in the named policy block. Set or policy references that occur as a result of an <b>apply</b> statement are not listed.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
	The <b>policy</b> keyword was changed to <b>route-policy</b> .

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl route-policy uses** command to display information about a specified named policy.

## Task ID

Task ID	Operations
route-policy	read

**Examples**

This example shows the following sample configuration:

```
prefix-set ten-net
 10.0.0.0/16 le 32
end-set
prefix-set too-specific
 0.0.0.0/0 ge 25 le 32
end-set
route-policy example-one
 if destination in ten-net then
  drop
 else
  set local-preference 200
  apply set-comms
 endif
end-policy
route-policy set-comms
 set community (10:1234) additive
end-policy
route-policy example-three
 if destination in too-specific then
  drop
 else
  apply example-one
  pass
 endif
end-policy
```

The following command lists the policies one and set-comms. It also lists the prefix sets too-specific and ten-net.

```
RP/0/0/CPU0:router# show rpl route-policy example-three uses all
```

```
Policies directly and indirectly applied by this policy:
```

```
-----
example-one set-comms
```

```
Sets referenced directly and indirectly
```

```
-----
(via applied policies) in this policy:
```

```
type prefix-set:
  ten-net too-specific
```

The sets example-one and set-comms are listed as policies that are used by the policy example-three. The policy example-one is listed because route policy example-three uses it in an **apply** statement. The policy set-comms is also listed because example-one applies it. Similarly, the prefix-set too-specific is used directly in the **if** statement in the policy example-three, and the prefix-set ten-net is used in the policy example-one. The optional **direct** keyword can be used to limit the output to just those sets and policies that are used within the example-three block itself, as shown in the following example:

```
RP/0/0/CPU0:router# show rpl route-policy example-three uses all direct
```

```
Policies directly applied by this policy:
```

```
-----
example-one
```

```
Sets used directly in this policy
```

```
-----
type prefix-set:
  too-specific
```

As can be seen in the output, the route policy set-comms and the prefix set ten-net are no longer included in the output when the **direct** keyword is used. The **direct** form of the command considers only those sets or

policies used in the specified route policy and any additional policies or sets that may be used if you follow the hierarchy of **apply** statements.

This table describes the significant fields shown in the display.

**Table 14: show rpl route-policy uses Field Descriptions**

Field	Description
type	Displays the type used in the policy configuration. Values for type are prefix-set, community-set, extcommunity-set, and as-path-set.

# show rpl unused as-path-set

To display the AS path sets that are defined but not used by a policy at an attach point or referenced in a policy using an **apply** statement, use the **show rpl unused as-path-set** command in EXEC mode.

## show rpl unused as-path-set [detail]

### Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for unused AS path sets.
---------------	---

### Command Default

No default behavior or values

### Command Modes

EXEC

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused as-path-set** command to display all AS path sets that are not used in a policy at an attach point either directly or indirectly and are not referenced by any policies in the system.

### Task ID

Task ID	Operations
route-policy	read

### Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
  route-policy policy_2 in
!
!
!
as-path-set as_path_set_ex1
  ios-regex '^_65500_$',
  ios-regex '^_65501_$'
end-set
!
as-path-set as_path_set_ex2
  ios-regex '^_65502_$',
  ios-regex '^_65503_$'
end-set
!
as-path-set as_path_set_ex3
  ios-regex '^_65504_$',
  ios-regex '^_65505_$'
end-set
!
route-policy sample
  if (destination in sample) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_5
  apply sample
  apply policy_3

```

```
end-policy
```

Given this sample configuration, the **show rpl unused as-path-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl unused as-path-set
```

```
ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced
```

```
The following as-path-sets are UNUSED
```

```
-----
as_path_set_ex3
```

## Related Commands

<a href="#">show rpl unused community-set, on page 273</a>	Displays the community sets that are not referenced at all.
<a href="#">show rpl unused extcommunity-set, on page 276</a>	Displays the extended community sets that are not referenced at all.
<a href="#">show rpl unused prefix-set, on page 278</a>	Displays the prefix sets that are not referenced at all.
<a href="#">show rpl unused rd-set, on page 281</a>	Displays the RD sets that are not referenced at all.
<a href="#">show rpl unused route-policy, on page 283</a>	Displays the route policies that are not referenced at all.



## show rpl unused community-set

To display the community sets that are defined but not used by a policy at an attach point or referenced in a policy using an **apply** statement, use the **show rpl unused community-set** command in EXEC mode.

**show rpl unused community-set [detail]**

### Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for unused community sets.
---------------	---

### Command Default

No default behavior or values

### Command Modes

EXEC

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused community-set** command to display all the community sets that are not used in a policy at an attach point either directly or indirectly and are not referenced by any policies in the system.

### Task ID

Task ID	Operations
route-policy	read

### Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
remote-as 12
address-family ipv4 unicast
route-policy policy_2 in
!
!
!

community-set comm_set_ex1
  65500:1,
  65500:2,
  65500:3
end-set
!
community-set comm_set_ex2
  65501:1,
  65501:2,
  65501:3
end-set
!
community-set comm_set_ex3
  65502:1,
  65502:2,
  65502:3
end-set
!
route-policy sample
  if (destination in sample) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
      set community (10:999) additive
    endif
  endif
end-policy
!

```

```
route-policy policy_5
  apply sample
  apply policy_3
end-policy
```

Given this sample configuration, the **show rpl unused community-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl unused community-set
```

```
ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced
```

```
The following community-sets are UNUSED
-----
comm_set_ex3
```

## Related Commands

Command	Description
<a href="#">show rpl unused as-path-set, on page 270</a>	Displays the AS path sets that are not referenced at all.
<a href="#">show rpl unused extcommunity-set, on page 276</a>	Displays the extended community sets that are not referenced at all.
<a href="#">show rpl unused prefix-set, on page 278</a>	Displays the prefix sets that are not referenced at all.
<a href="#">show rpl unused rd-set, on page 281</a>	Displays the RD sets that are not referenced at all.
<a href="#">show rpl unused route-policy, on page 283</a>	Displays the route policies that are not referenced at all.

# show rpl unused extcommunity-set

To display the extended community sets that are defined but not used by a policy at an attach point or referenced in a policy using an **apply** statement, use the **show rpl unused extcommunity-set** command in EXEC mode.

**show rpl unused extcommunity-set** [**cost**| **detail**| **rt**| **soo**]

## Syntax Description

<b>cost</b>	(Optional) Displays the unused extended-community cost objects.
<b>rt</b>	(Optional) Displays the unused extended community RT objects.
<b>soo</b>	(Optional) Displays the unused extended-community SoO objects.
<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for unused extended community sets.

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The following keywords were added: <ul style="list-style-type: none"> <li>• <b>cost</b></li> <li>• <b>rt</b></li> <li>• <b>soo</b></li> <li>• <b>detail</b> .</li> </ul>

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused extcommunity-set** command to display all extended community sets that are not used in a policy at an attach point either directly or indirectly and are not referenced by any policies in the system.

**Task ID**

Task ID	Operations
route-policy	read

**Examples**

The following is sample output for the **show rpl unused extcommunity-set** command:

```
RP/0/0/CPU0:router:router# show rpl unused extcommunity-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following extcommunity-sets are UNUSED
-----
ext_comm_set_ex3
```

**Related Commands**

Command	Description
<a href="#">show rpl unused as-path-set, on page 270</a>	Displays the AS path sets that are not referenced at all.
<a href="#">show rpl unused community-set, on page 273</a>	Displays community sets that are not referenced at all.
<a href="#">show rpl unused prefix-set, on page 278</a>	Displays prefix sets that are not referenced at all.
<a href="#">show rpl unused rd-set, on page 281</a>	Displays the RD sets that are not referenced at all.
<a href="#">show rpl unused route-policy, on page 283</a>	Displays the route policies that are not referenced at all.

# show rpl unused prefix-set

To display the prefix sets that are defined but not used by a policy at an attach point or referenced in a policy using an **apply** statement, use the **show rpl unused prefix-set** command in EXEC mode.

**show rpl unused prefix-set [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for unused prefix sets.
---------------	--

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.3.0	The <b>detail</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused prefix-set** command to display all prefix sets that are not used in a policy at an attach point either directly or indirectly and are not referenced by any policies in the system.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
router bgp 2
 address-family ipv4 unicast
 !
 neighbor 10.0.101.2
  remote-as 100
  address-family ipv4 unicast
   route-policy policy_1 in
 !
```

```

!
neighbor 10.0.101.3
  remote-as 12
  address-family ipv4 unicast
  route-policy policy_2 in
!
!
!

prefix-set sample
  0.0.0.0/0,
  0.0.0.0/0 ge 25 le 32,
  10.0.0.0/8 ge 8 le 32,
  192.168.0.0/16 ge 16 le 32,
  224.0.0.0/20 ge 20 le 32,
  240.0.0.0/20 ge 20 le 32
end-set
!
prefix-set prefix_set_ex1
  10.0.0.0/16 ge 16 le 32,
  0.0.0.0/0 ge 25 le 32,
  0.0.0.0/0
end-set
!
prefix-set prefix_set_ex2
  220.220.220.0/24 ge 24 le 32,
  220.220.120.0/24 ge 24 le 32,
  220.220.130.0/24 ge 24 le 32
end-set
!
prefix-set prefix_set_ex3
  221.221.220.0/24 ge 24 le 32,
  221.221.120.0/24 ge 24 le 32,
  221.221.130.0/24 ge 24 le 32
end-set
!
route-policy sample
  if (destination in sample) then
    drop
  endif
end-policy
!
route-policy policy_1
  if (destination in prefix_set_ex1) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex1) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_2
  if (destination in prefix_set_ex1) then
    if (community matches-any comm_set_ex1) then
      set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex1) then
      set community (10:999) additive
    endif
  endif
end-policy
!
route-policy policy_3
  if (destination in prefix_set_ex2) then
    set local-preference 100
  endif
  if (as-path in as_path_set_ex2) then
    set community (10:333) additive
  endif
end-policy
!
route-policy policy_4
  if (destination in prefix_set_ex2) then

```

```

    if (community matches-any comm_set_ex2) then
        set community (10:666) additive
    endif
    if (extcommunity matches-any ext_comm_set_rt_ex2) then
        set community (10:999) additive
    endif
endif
end-policy
!
route-policy policy_5
    apply sample
    apply policy_3
end-policy
-----
ext_comm_set_ex3

```

Given this sample configuration, the **show rpl unused prefix-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl unused prefix-set
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

```

The following prefix-sets are UNUSED
-----
prefix_set_ex3

```

## Related Commands

Command	Description
<a href="#">show rpl unused as-path-set, on page 270</a>	Displays AS path sets that are not referenced at all.
<a href="#">show rpl unused community-set, on page 273</a>	Displays community sets that are not referenced at all.
<a href="#">show rpl unused extcommunity-set, on page 276</a>	Displays extended community sets that are not referenced at all.
<a href="#">show rpl unused rd-set, on page 281</a>	Displays the RD sets that are not referenced at all.
<a href="#">show rpl unused route-policy, on page 283</a>	Displays the route policies that are not referenced at all.



## show rpl unused rd-set

To display the route distinguisher (RD) sets that are defined but not used by a policy at an attach point or referenced in a policy using an **apply** statement, use the **show rpl unused rd-set** command in EXEC mode.

**show rpl unused rd-set [detail]**

<b>Syntax Description</b>	<b>detail</b> (Optional) Displays the content of the object and all referenced objects for unused RD sets.
---------------------------	--

<b>Command Default</b>	No default behavior or values
------------------------	-------------------------------

<b>Command Modes</b>	EXEC
----------------------	------

<b>Command History</b>	Release	Modification
	Release 3.4.0	This command was introduced.

**Usage Guidelines**

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused rd-set** command to display all of the RD sets that are not used in a policy at an attach point either directly or indirectly and are not referenced by any policies in the system.

<b>Task ID</b>	Task ID	Operations
	route-policy	read

**Examples**

The **show rpl unused rd-set** command displays the following information:

```
RP/0/0/CPU0:router# show rpl unused rd-set

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following rd-sets are UNUSED
-----
None found with this status.
```

**Related Commands**

Command	Description
<a href="#">show rpl unused as-path-set, on page 270</a>	Displays the AS path sets that are not referenced at all.
<a href="#">show rpl unused community-set, on page 273</a>	Displays the community sets that are not referenced at all.
<a href="#">show rpl unused extcommunity-set, on page 276</a>	Displays the extended community sets that are not referenced at all.
<a href="#">show rpl unused prefix-set, on page 278</a>	Displays the prefix sets that are not referenced at all.
<a href="#">show rpl unused route-policy, on page 283</a>	Displays the route policies that are not referenced at all.

# show rpl unused route-policy

To display the route policies that are defined but not used at an attach point or referenced using an **apply** statement, use the **show rpl unused route-policy** command in EXEC mode.

**show rpl unused route-policy [detail]**

## Syntax Description

<b>detail</b>	(Optional) Displays the content of the object and all referenced objects for unused route policies.
---------------	---

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

Release	Modification
Release 3.2	This command was introduced. The <b>policy</b> keyword was changed to <b>route-policy</b> .
Release 3.3.0	The <b>detail</b> keyword was added.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **show rpl unused route-policy** command to display route policies that are defined but not used at an attach point or referenced from another policy using an **apply** statement.

## Task ID

Task ID	Operations
route-policy	read

## Examples

This example shows the following sample configuration:

```
RP/0/0/CPU0:router# show run | begin prefix-set
Building configuration...
prefix-set prefix_set_ex1
  10.0.0.0/16 ge I6 le 32,
  0.0.0.0/0 ge 25 le 32,
```

```

    0.0.0.0/0
end-set
!
prefix-set prefix_set_ex2
    220.220.220.0/24 ge 24 le 32,
    220.220.120.0/24 ge 24 le 32,
    220.220.130.0/24 ge 24 le 32
end-set
!
as-path-set as_path_set_ex1
    ios-regex '^_65500_$',
    ios-regex '^_65501_$'
end-set
!
as-path-set as_path_set_ex2
    ios-regex '^_65502_$',
    ios-regex '^_65503_$'
end-set
!
as-path-set as_path_set_ex3
    ios-regex '^_65504_$',
    ios-regex '^_65505_$'
end-set
!
community-set comm_set_ex1
    65500:1,
    65500:2,
    65500:3
end-set
!
community-set comm_set_ex2
    65501:1,
    65501:2,
    65501:3
end-set
!
extcommunity-set rt_ext_comm_set_rt_ex1
    1.2.3.4:34
end-set
!
extcommunity-set rt_ext_comm_set_rt_ex2
    2.3.4.5:36
end-set
!
route-policy sample
    if (destination in sample) then
        drop
    endif
end-policy
!
route-policy policy_1
    if (destination in prefix_set_ex1) then
        set local-preference 100
    endif
    if (as-path in as_path_set_ex1) then
        set community (10:333) additive
    endif
end-policy
!
route-policy policy_2
    if (destination in prefix_set_ex1) then
        if (community matches-any comm_set_ex1) then
            set community (10:666) additive
        endif
        if (extcommunity rt matches-any ext_comm_set_rt_ex1) then
            set community (10:999) additive
        endif
    endif
end-policy
!
route-policy policy_3
    if (destination in prefix_set_ex2) then
        set local-preference 100

```

```

endif
if (as-path in as_path_set_ex2) then
    set community (10:333) additive
endif
end-policy
!
route-policy policy_4
if (destination in prefix_set_ex2) then
    if (community matches-any comm_set_ex2) then
        set community (10:666) additive
    endif
    if (extcommunity rt matches-any ext_comm_set_rt_ex2) then
        set community (10:999) additive
    endif
endif
end-policy
!
route-policy policy_5
    apply sample
    apply policy_3
end-policy
!
route ipv4 0.0.0.0/0 10.91.37.129
route ipv4 10.91.36.0/23 10.91.37.129
route ipv4 10.91.38.0/24 10.91.37.129
end

```

In the following example, route policies that are defined but not used at an attach point or referenced from another policy using an **apply** statement are displayed using the **show rpl unused route-policy** command.

```
RP/0/0/CPU0:router# show rpl unused route-policy
```

```

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

```

The following policies are (UNUSED)

```

-----
policy_1
policy_2
policy_4
policy_5

```

## Related Commands

Command	Description
<a href="#">show rpl unused as-path-set, on page 270</a>	Displays AS path sets that are not referenced at all.
<a href="#">show rpl unused community-set, on page 273</a>	Displays community sets that are not referenced at all.
<a href="#">show rpl unused extcommunity-set, on page 276</a>	Displays extended community sets that are not referenced at all.
<a href="#">show rpl unused prefix-set, on page 278</a>	Displays prefix sets that are not referenced at all.
<a href="#">show rpl unused rd-set, on page 281</a>	Displays the RD sets that are not referenced at all.

## source in

To test the source of a Border Gateway Protocol (BGP) route against the address contained in either a named or an inline prefix set, use the **source in** command in route-policy configuration mode.

**source in** {*prefix-set-name*| *inline-prefix-set*| *parameter*}

### Syntax Description

<i>prefix-set-name</i>	Name of a prefix set.
<i>inline-prefix-set</i>	Inline prefix set. The inline prefix set must be enclosed in parentheses.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

No default behavior or values

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.4.0	The <i>parameter</i> argument was added.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **source in** command as a conditional expression within an **if** statement to test the source of the route against the data in either a named or an inline prefix set. A comparison that references a prefix set with zero elements in it returns false.



#### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

The source of a BGP route is the IP peering address of the neighboring router from which the route was received.

The prefix set can contain both IPv4 and IPv6 prefix specifications.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, the source of a BGP route is tested against the data in the prefix set my-prefix-set:

```
RP/0/0/CPU0:router(config)# route-policy policy-A
RP/0/0/CPU0:router(config-rpl)# if source in my-prefix-set then
```

In this example, the source of a BGP route is tested against the data in an inline IPv4 prefix set:

```
RP/0/0/CPU0:router(config)# route-policy policy-B
RP/0/0/CPU0:router(config-rpl)# if source in (10.0.0.8, 10.0.0.20) then
```

In this example, the source of a route is tested against the data in an inline IPv6 prefix set:

```
RP/0/0/CPU0:router(config)# route-policy policy-C
RP/0/0/CPU0:router(config-rpl)# if source in (2001:0:0:1::/64, 2001:0:0:2::/64) then
```

**Related Commands**

Command	Description
<a href="#">prefix-set, on page 104</a>	Enters a prefix set configuration mode and defines a prefix set.

# suppress-route

To indicate that a given component of a BGP aggregate should be suppressed, use the **suppress-route** command in route-policy configuration mode.

**suppress-route**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **suppress-route** command to indicate that a given component of an aggregate should be suppressed, that is, not advertised by BGP. See the [unsuppress-route](#), on page 296 command for information on overriding the **suppress-route** command for individual neighbors.

The **suppress-route** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, if the destination is in 10.1.0.0/16, then the route is not advertised:

```
RP/0/0/CPU0:router(config)# route-policy check-aggregator
RP/0/0/CPU0:router(config-rpl)# if destination in (10.1.0.0/16) then
RP/0/0/CPU0:router(config-rpl-if)# suppress-route

RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl-if)# end-policy
```



**Related Commands**

Command	Description
<a href="#">unsuppress-route</a> , on page 296	Indicates that a given component of an aggregate should be unsuppressed

# tag

To match a specific tag value, use the **tag** command in route-policy configuration mode.

**tag** {eq| ge| le| is} {integer| parameter}

## Syntax Description

<b>eq   ge   le   is</b>	Equal to; greater than or equal to; less than or equal to.
<i>integer</i>	Integer value. Range is 0 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.2	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **tag** command as a conditional expression within an **if** statement to match a specific tag value.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

A tag is a 32-bit integer that can be associated with a given route within the RIB.

The **eq** operator matches either a specific tag value or a parameter value. Its variants **ge** and **le** match a range of tag values that are either greater than or equal to or less than or equal to the supplied value or parameter.

## Task ID

Task ID	Operations
route-policy	read, write

### Examples

In the following example, if the tag equals 10, then the condition returns true:

```
RP/0/RSP0RP0/CPU0:router(config-rpl)# if tag eq 10 then
```

# tag in

To match a tag entry in a named tag set or inline tag set, use the **tag in** command in route-policy configuration mode.

**tag in** {*tag-set-name*| *inline-tag-set*| *parameter*}

## Syntax Description

<i>tag-set-name</i>	Name of a tag set. The tag-set accepts 32-bit Integer value. Range is 0 to 4294967295.
<i>inline-tag-set</i>	Inline tag set. The inline tag set must be enclosed in parentheses.
<i>parameter</i> parameter	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 4.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **tag in** command as a conditional expression within an **if** statement to match a tag entry in a named tag set or inline tag set.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command takes either a named tag set or an inline tag set value as an argument. The condition returns true if the tag entry matches any entry in the tag set or inline tag set. An attempt to match a tag using a tag set that is defined but contains no elements returns false.

The routing policy language (RPL) provides the ability to test tags for a match to a list of tag match specifications using the **in** operator. The **tag in** command is protocol-independent.

**Task ID**

Task ID	Operations
route-policy	read, write

**Examples**

In the following example, a tag set named `my-tag-set` is defined and a route policy named `use-tag-in` is created. Within the `use-tag-in` route policy, the `tag in` command is used within an `if` statement to learn if the tag is in the tag-set named `my-tag-set`. If it is, then local preference is set to 100. If it is not in `my-tag-set` but does match the next tag specifications, then local preference is set to 200.

```
RP/0/0/CPU0:router(config)#tag-set my-tag-set
RP/0/0/CPU0:router(config-tag)#1000
RP/0/0/CPU0:router(config-tag)#3000
RP/0/0/CPU0:router(config-tag)#end-set

RP/0/0/CPU0:router(config)#route-policy use-tag-in
RP/0/0/CPU0:router(config-rpl)#if tag in my-tag-set then
RP/0/0/CPU0:router(config-rpl-if)#set local-preference 100
RP/0/0/CPU0:router(config-rpl-if)#elseif tag in (2000, 4000) then
RP/0/0/CPU0:router(config-rpl-elseif)#set local-preference 200
RP/0/0/CPU0:router(config-rpl-elseif)#endif
RP/0/0/CPU0:router(config-rpl)#end policy
```

# tag-set

To enter tag set configuration mode and define a tag set, use the **tag-set** command in global configuration mode. To remove a named tag set, use the **no** form of this command.

**tag-set** *name*

**no tag-set** *name*

## Syntax Description

<i>name</i>	Name of a tag set.
-------------	--------------------

## Command Default

None

## Command Modes

Global configuration

## Command History

Release	Modification
Release 4.3.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **tag-set** command to enter tag set configuration mode and define a tag set. A tag-set is a 32-bit integer that can be associated with a given route within the RIB.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, a tag set named `my-tag-set` is defined and a route policy named `use-tag-in` is created. Within the `use-tag-in` route policy, the **tag in** command is used within an **if** statement to learn if the tag is in the tag-set named `my-tag-set`. If it is, then local preference is set to 100. If it is not in `my-tag-set` but does match the next tag specifications, then local preference is set to 200.

```
RP/0/0/CPU0:router(config)#tag-set my-tag-set
RP/0/0/CPU0:router(config-tag)#1000
RP/0/0/CPU0:router(config-tag)#3000
RP/0/0/CPU0:router(config-tag)#end-set
```

```
RP/0/0/CPU0:router(config)#route-policy use-tag-in  
RP/0/0/CPU0:router(config-rpl)#if tag in my-tag-set then  
RP/0/0/CPU0:router(config-rpl-if)#set local-preference 100  
RP/0/0/CPU0:router(config-rpl-if)#elseif tag in (2000, 4000) then  
RP/0/0/CPU0:router(config-rpl-elseif)#set local-preference 200  
RP/0/0/CPU0:router(config-rpl-elseif)#endif  
RP/0/0/CPU0:router(config-rpl)#end policy
```

# unsuppress-route

To indicate that a given component of a BGP aggregate should be unsuppressed, use the **unsuppress-route** command in route-policy configuration mode.

**unsuppress-route**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values

**Command Modes** Route-policy configuration

Command History	Release	Modification
	Release 3.2	This command was introduced.

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **unsuppress-route** command to indicate that a given component of an aggregate should be unsuppressed, that is, allowed to be advertised by BGP again. This command affects routes that have been suppressed in the generation of BGP aggregates. If the request to unsuppress a route is encountered in a policy at a neighbor-out attach point, it guarantees that the routes that it affects are advertised to that neighbor even if that route was suppressed using the **suppress-route** command in a policy at the aggregation attach point.

The **unsuppress-route** command can be used as an action statement within an **if** statement. For a list of all action statements available within an **if** statement, see the **if** command.

Task ID	Task ID	Operations
	route-policy	read, write

**Examples** In the following example, if the destination is in 10.1.0.0/16, then the route is not advertised:

```
RP/0/0/CPU0:router(config)# route-policy check-aggregate
RP/0/0/CPU0:router(config-rpl)# if destination in (10.1.0.0/16) then
RP/0/0/CPU0:router(config-rpl-if)# unsuppress-route

RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)# end-policy
```



Assuming that the policy is attached at a neighbor-out attach point, if the route 10.1.0.0/16 was suppressed in a policy at an aggregation attach point, 10.1.0.0/16 is advertised to the neighbor. Routes continue to be suppressed in advertisements to other BGP neighbors unless a specific policy is attached to unsuppress the route.

**Related Commands**

Command	Description
<a href="#">suppress-route</a> , <a href="#">on page 288</a>	Indicates that a given component of a BGP aggregate should be suppressed.

## var globalVarN

To assign a value to route-policy global variable "globalVar1, globalVar2, globalVar3, globalVar4, and globalVar5", use the **var globalVarN** command in route-policy configuration mode.

**var globalVarN** {*number* | *parameter*}

### Syntax Description

<b>number</b>	Value assigned to a 32-bit unsigned integer. Range is from 1 to 4294967295.
<b>parameter</b>	Parameter name. The parameter name must be preceded with a "\$."

### Command Default

If the var globalVarN statement is not present then the value for globalVarN is zero.

### Command Modes

Route-policy configuration

### Command History

Release	Modification
Release 5.1.3	This command was introduced.

### Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

The globalVarN variables can be used to control policy execution flow. You can assign a weightage to mark the flow and check the value using an if statement.



#### Note

The var globalVarN represents statements "var globalVar1" , "var globalVar2" , "var globalVar3" , "var globalVar4" and "var globalVar5". This command can be used as an assignment statement within an **if** statement or in child policies and the value can be used to compare in an **if** statement.

### Task ID

Task ID	Operation
route-policy	read, write

## Examples

In the following example, the globalVar1 is set to 123 and globalVar2 is set to the value stored in parameter \$param.

```
RP/0/0/CPU0:ios(config-rpl)#var globalVar1 123
RP/0/0/CPU0:ios(config-rpl)#var globalVar2 $param
```

# vpn-distinguisher is

To match a specific Border Gateway Protocol (BGP) VPN distinguisher, use the **vpn-distinguisher is** command in route-policy configuration mode.

**vpn-distinguisher is** {*number*|*parameter*}

## Syntax Description

<i>number</i>	Value assigned to a 32-bit unsigned integer. Range is from 1 to 4294967295.
<i>parameter</i>	Parameter name. The parameter name must be preceded with a "\$."

## Command Default

No default behavior or values

## Command Modes

Route-policy configuration

## Command History

Release	Modification
Release 3.4.0	This command was introduced.

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **vpn-distinguisher is** command as a conditional expression within an **if** statement to test the value of the origin attribute.

A VPN distinguisher is used in Layer 3 VPN networks for enhanced individual VPN control and to avoid route target mapping at AS boundaries in inter-AS VPN networks. Route target extended communities are removed at neighbor outbound and the VPN distinguisher value is applied on the BGP route as an extended community. When the route is received on a neighboring router in another AS, the VPN distinguisher is removed and mapped to a route target extended community.



### Note

For a list of all conditional expressions available within an **if** statement, see the **if** command.

This command can be parameterized.

## Task ID

Task ID	Operations
route-policy	read, write

## Examples

In the following example, the origin is tested within an **if** statement to learn if it is either **igp** or **egp** :

```
RP/0/0/CPU0:router(config-rpl)# if origin is igp or origin is egp then
```

In the following example, a parameter is used to match a specific origin type:

```
RP/0/0/CPU0:router(config)# route-policy bar($origin)
RP/0/0/CPU0:router(config-rpl)# if origin is $origin then
RP/0/0/CPU0:router(config-rpl-if)# set med 20
RP/0/0/CPU0:router(config-rpl-if)# endif
RP/0/0/CPU0:router(config-rpl)#
```

vpn-distinguisher is