

RPL Commands

- as-path in, on page 3
- as-path-set, on page 5
- community matches-any, on page 7
- community matches-every, on page 9
- community-set, on page 11
- destination in, on page 14
- done, on page 16
- drop, on page 18
- drop (ePBR), on page 19
- end-policy, on page 20
- end-set, on page 21
- if, on page 23
- local-preference, on page 28
- med, on page 29
- next-hop-type is, on page 30
- pass, on page 31
- prefix-set, on page 32
- rib-has-route, on page 34
- route-policy (RPL), on page 35
- set community, on page 37
- set extcommunity transitive-bandwidth, on page 39
- set local-preference, on page 40
- set med, on page 41
- set next-hop, on page 43
- set origin, on page 45
- show rpl, on page 46
- show rpl active as-path-set, on page 48
- show rpl active community-set, on page 50
- show rpl active extcommunity-set, on page 52
- show rpl active prefix-set, on page 55
- show rpl active rd-set, on page 57
- show rpl active route-policy, on page 59
- show rpl as-path-set, on page 61

- show rpl as-path-set attachpoints, on page 62
- show rpl as-path-set references, on page 64
- show rpl community-set, on page 66
- show rpl community-set attachpoints, on page 68
- show rpl community-set references, on page 70
- show rpl extcommunity-set, on page 72
- show rpl inactive as-path-set, on page 75
- show rpl inactive community-set, on page 77
- show rpl inactive extcommunity-set, on page 79
- show rpl inactive prefix-set, on page 81
- show rpl inactive rd-set, on page 83
- show rpl inactive route-policy, on page 85
- show rpl maximum, on page 87
- show rpl policy-global references, on page 89
- show rpl prefix-set, on page 91
- show rpl prefix-set attachpoints, on page 92
- show rpl prefix-set references, on page 94
- show rpl rd-set, on page 96
- show rpl rd-set attachpoints, on page 97
- show rpl rd-set references, on page 99
- show rpl route-policy, on page 101
- show rpl route-policy attachpoints, on page 104
- show rpl route-policy inline, on page 106
- show rpl route-policy references, on page 108
- show rpl route-policy uses, on page 111
- show rpl unused as-path-set, on page 114
- show rpl unused community-set, on page 117
- show rpl unused extcommunity-set, on page 120
- show rpl unused prefix-set, on page 121
- show rpl unused rd-set, on page 124
- show rpl unused route-policy, on page 125
- transmit (ePBR), on page 128

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

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

Command Default

No default behavior or values

Command Modes

Route-policy configuration

Usage Guidelines

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:

```
Router(config)# as-path-set my-as-set
Router(config-as)# ios-regex '_12$',
Router(config-as)# ios-regex '_13$'
Router(config-as)# end-set
```

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

```
Router(config-rpl)# if as-path in my-as-set then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# endif
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:

```
Router(config-rpl)# if as-path in (ios-regex '_12$,ios-regex '_13$') then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# endif
Router(config-rpl)#
```

as-path-set

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

as-path-set *name*

Syntax Description

name	Name of the AS path set.

Command Default

No default behavior or values

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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 any of the below option to specify an operation.

Options	Description
dfa-regex	Indicates the DFA (deterministic finite automata) style regular expression. It performs better for complex regular expressions. Single quotation marks are required around the regular expression.
ios-regex	Indicates the traditional IOS style regular expression. It performs better with simpler regular expressions. Single quotation marks are required around the regular expression.
length	Indicates the number of ASN (Autonomous System Number) in the AS path of a Border Gateway Protocol (BGP) route.
neighbor-is	Indicates the neighbor's AS-path number that can be matched with.
originates-from	Indicates the BGP AS from which the route originated.
passes-through	Indicates if the supplied integer or parameter appears anywhere in the AS path, or if the supplied sequence of integers and parameters appear, in the same order, anywhere in the AS path.

Options	Description
unique-length	Indicates the length of BGP AS-path, ignoring duplicates.

The above options can also be used as an inline set in a parenthesized list of comma-separated expressions.

Task ID

Task ID	Operations
route-policy	read, write

Examples

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

```
RP/0/RP0/CPU0:router(config)# as-path-set aset1
RP/0/RP0/CPU0:router(config-as)# ios-regex '_42$',
RP/0/RP0/CPU0:router(config-as)# ios-regex '_127$'
RP/0/RP0/CPU0:router(config-as)# end-set
```

The following is a sample of the as-path options used as an inline set.

```
RP/0/RP0/CPU0:router(config-rpl)# if as-path in (ios-regex '_42$', ios-regex$ '_127$')
RP/0/RP0/CPU0:router(config-rpl-if)# pass
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

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

community-set-name	Name of a community set.
inline-community-set	Inline community set. The inline community set must be enclosed in parentheses.
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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config)# community-set my-community-set
  RP/0/RP0/CPU0:router(config-comm)# 10:20,
  RP/0/RP0/CPU0:router(config-comm)# 10:30,
  RP/0/RP0/CPU0:router(config-comm)# 10:40
  RP/0/RP0/CPU0:router(config-comm)# end-set

  RP/0/RP0/CPU0:router(config)# route-policy community-matches-any-example
  RP/0/RP0/CPU0:router(config-rpl)# if community matches-any my-community-set then
  RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 100
  RP/0/RP0/CPU0:router(config-rpl-if)# elseif community matches-any ([10..25]:35) then
  RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 200
  RP/0/RP0/CPU0:router(config-rpl-elseif)# elseif community matches-any (30:[100..500])
then
  RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 300
  RP/0/RP0/CPU0:router(config-rpl-elseif)# endif
  RP/0/RP0/CPU0:router(config-rpl)# end-policy
```

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

community-set-name	Name of a community set.
inline-community-set	Inline community set. The inline community set must be enclosed in parentheses.
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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config)# community-set my-community-set
RP/0/RP0/CPU0:router(config-comm)# 10:20,
RP/0/RP0/CPU0:router(config-comm)# 10:30,
RP/0/RP0/CPU0:router(config-comm)# 10:40
RP/0/RP0/CPU0:router(config-comm)# end-set

RP/0/RP0/CPU0:router(config)# route-policy community-matches-every-example
RP/0/RP0/CPU0:router(config-rpl)# if community matches-every my-community-set then
RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/RP0/CPU0:router(config-rp-elseif)# elseif community matches-every (ios-regex
'_10:[0-9]0_') then
RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 200
RP/0/RP0/CPU0:router(config-rpl-elseif)# elseif community matches-every
(ios-regex'_20:[0-9]0_') then
RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 300
RP/0/RP0/CPU0:router(config-rpl-elseif)# endif
RP/0/RP0/CPU0:router(config-rpl)# end-policy
```

community-set

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

community-set *name*

Syntax Description

name	Name of the community set.

Command Default

No default behavior or values

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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
#-remark	Remark beginning with '#'
*****	Wildcard (any community or part thereof)
0-65535	16-bit half-community number
[	Left bracket to begin range
accept-own	Accept-Own (BGP well-known community)

Format	Description
dfa-regex	DFA (deterministic finite automata) style regular expression
internet	Internet (BGP well-known community)
ios-regex	Traditional IOS style regular expression
local-AS	Do not send outside local AS (BGP well-known community)
no-advertise	Do not advertise to any peer (BGP well-known community)
no-export	Do not export to next AS (BGP well-known community)
private-as	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/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#community-set cset_accept_own
RP/0/RP0/CPU0:router(config-comm)#accept-own
RP/0/RP0/CPU0:router(config-comm)#end-set
```

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

```
RP/0/RP0/CPU0:router(config)# community-set cset1
RP/0/RP0/CPU0:router(config-comm)# 12:34,
RP/0/RP0/CPU0:router(config-comm)# 12:56,
RP/0/RP0/CPU0:router(config-comm)# 12:78,
RP/0/RP0/CPU0:router(config-comm)# internet
RP/0/RP0/CPU0:router(config-comm)# end-set
```

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

```
RP/0/RP0/CPU0:router(config)# community-set cset2
RP/0/RP0/CPU0:router(config-comm)# 123:456,
RP/0/RP0/CPU0:router(config-comm)# no-advertise,
```

```
RP/0/RP0/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/RP0/CPU0:router(config)# community-set cset3
RP/0/RP0/CPU0:router(config-comm)# 123:*
RP/0/RP0/CPU0:router(config-comm)# end-set
```

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

prefix-set-name	Name of a prefix set.
inline-prefix-set	Inline prefix set. The inline prefix set must be enclosed in parentheses.
parameter 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 7.0.12	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 taskIDs. 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.
- 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.
- Use the **destination-prefix in** command to filter a flowspec route in BGP, and the **match destination-address** command to create a flowspec rule that matches a specific destination.
- Use the **destination in** command to filter BGP routes for other address families, and use the **destination-prefix in** command to filter flowspec routes in BGP that include a **match destination-address** clause.
- 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/RP0/CPU0:router(config)# prefix-set my-prefix-set
  RP/0/RP0/CPU0:router(config-pfx)# 10.0.0.1/32,
  RP/0/RP0/CPU0:router(config-pfx)# fe80::203:0:0:0/64,
  RP/0/RP0/CPU0:router(config-pfx)# 10.0.0.2/24 le 32
  RP/0/RP0/CPU0:router(config-pfx)# end-set

  RP/0/RP0/CPU0:router(config)# route-policy use-destination-in
  RP/0/RP0/CPU0:router(config-rpl)# if destination in my-prefix-set then
  RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 100
  RP/0/RP0/CPU0:router(config-rpl-if)# elseif destination in (10.0.0.1/32, 10.0.0.2/24 le
32) then
  RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 200
  RP/0/RP0/CPU0:router(config-rpl-elseif)# endif
  RP/0/RP0/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/RP0/CPU0:router(config)# prefix-set ipv6-prefix-set
  RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:1::/64,
  RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:2::/64,
  RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:3::/64,
  RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:4::/64
  RP/0/RP0/CPU0:router(config-pfx)# end-set

  RP/0/RP0/CPU0:router(config)# route-policy ipv6-destination-in
  RP/0/RP0/CPU0:router(config-rpl)# if destination in ipv6-prefix-set then
  RP/0/RP0/CPU0:router(config-rpl-if)# set next-hop 2001:abcd:fedc::1
  RP/0/RP0/CPU0:router(config-rpl-if)# elseif destination in (2001::1, 2002:1:2:3::/64)
then
  RP/0/RP0/CPU0:router(config-rpl-elseif)# set next-hop
1111:2222:3333:4444:5555:6666:7777:8888
  RP/0/RP0/CPU0:router(config-rpl-elseif)# endif
  RP/0/RP0/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

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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

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 in encounters the *done* statement.

```
RP/0/RP0/CPU0:router(config)# route-policy done_st_example
RP/0/RP0/CPU0:router(config-rpl)# if destination in (29.0.0.0/8 le 32) then
RP/0/RP0/CPU0:router(config-rpl-if)# set community 102:12
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)# if destination in (39.0.0.0/8 le 32) then
RP/0/RP0/CPU0:router(config-rpl-if)# set community 102:39
```

```
RP/0/RP0/CPU0:router(config-rpl-if)# done
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)# if destination in (49.0.0.0/8 le 32) then
RP/0/RP0/CPU0:router(config-rpl-if)# set community 102:49
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)# if destination in (59.0.0.0/8 le 32) then
RP/0/RP0/CPU0:router(config-rpl-if)# set community 102:59
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/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

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines 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/RP0/CPU0:router(config-rpl)# if destination in pset1 then
RP/0/RP0/CPU0:router(config-rpl-if)# drop
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

drop (ePBR)

The command discards packets that match the associated class map criteria when configured under a PBR class within a policy map.

drop

Command Default

None.

Command Modes

Policy map configuration mode.

Command History

Release	Modification
Release 25.3.1	This command was introduced.

Task ID

Task ID	Operation
qos	read, write

```
Router(config)#policy-map type pbr INGRESS-SECURITY-POLICY
Router(config-pmap)#class type traffic UNWANTED-SOURCE-TRAFFIC
Router(config-pmap-c)#drop
Router(config-pmap-c)#exit
```

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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config)#route-policy med-to-local-pref
RP/0/RP0/CPU0:router(config-rpl)#if med eq 150 then
RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 10
RP/0/RP0/CPU0:router(config-rpl-if)# elseif med eq 200 then
RP/0/RP0/CPU0:router(config-elseif)# set local-preference 60
RP/0/RP0/CPU0:router(config-elseif)# elseif med eq 250 then
RP/0/RP0/CPU0:router(config-elseif)# set local-preference 0

RP/0/RP0/CPU0:router(config-elseif)# endif
RP/0/RP0/CPU0:router(config-rpl)# end-policy
```

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 XR Config 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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config)# as-path-set aset1
RP/0/RP0/CPU0:router(config-as)# ios-regex '_42$',
RP/0/RP0/CPU0:router(config-as)# ios-regex '_127$'

RP/0/RP0/CPU0:router(config-as)# end-set
RP/0/RP0/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/RP0/CPU0:router(config)# rd-set my_rd_set
RP/0/RP0/CPU0:router(config-rd)# 172.16.0.0/16:*,
RP/0/RP0/CPU0:router(config-rd)# 172.17.0.0/16:100,
RP/0/RP0/CPU0:router(config-rd)# 192:*,
RP/0/RP0/CPU0:router(config-rd)# 192:100
```

```
RP/0/RP0/CPU0:router(config-rd)# end-set
```

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

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

Command Default

No default behavior or values

Command Modes

Route-policy configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **if** command uses a conditional expression to decide which actions or dispositions should be taken for a given route. Table 1 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 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
as-path in	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.

Command	Description
as-path is-local	Determines if the router (or another router within this autonomous system or confederation) originated the route.
as-path length	Performs a conditional check based on the length of the AS path.
as-path neighbor-is	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.
as-path originates-from	Tests an AS path against the AS sequence beginning with the AS number that originated a route.
as-path passes-through	Tests to learn if the specified integer or parameter appears anywhere in the AS path or if the sequence of integers and parameters appears.
as-path unique-length	Performs specific checks based on the length of the AS path.
community is-empty	Learns if a route has community attributes associated with it.
community matches-any	Matches any element of a community set.
community matches-every	Matches every element of a community set.
destination in	Matches a destination entry in a named prefix set or inline prefix set.
extcommunity rt is-empty	Learns if a route has RT extended community attributes associated with it.
extcommunity rt matches-any	Matches elements of an RT extended community set.
extcommunity rt matches-every	Matches every element of an RT extended community set.
extcommunity rt matches-within	Matches at least one element of a Border Gateway Protocol (BGP) route target (RT) extended community set.
extcommunity soo is-empty	Learns if a route has SoO extended community attributes associated with it.
extcommunity soo matches-any	Matches elements of an SoO extended community set.
extcommunity soo matches-every	Matches every element of an SoO extended community set.
local-preference	Specifies BGP local-preference attribute
med	Compares the MED to an integer value or a parameterized value.
next-hop in	Compares the next-hop associated with the route to data contained in either a named or an inline prefix set.
orf prefix in	Matches a prefix in a prefix set or an inline prefix set.
origin is	Tests the value of the origin attribute.
path-type is	Tests the path type.

Command	Description
protocol	Checks if a protocol is installing the route.
rd in	Compares the RD associated with the route to data contained in either a named or an inline RD set.
rib-has-route	Checks if a route is in the RIB.
route-has-label	Checks if a route has a Multiprotocol Label Switching (MPLS) label.
route-type is	Compares route types when redistribution is being performed into BGP, OSPF, or IS-IS.
source in	Tests the source of the route against the data in either a named or an inline prefix set.
tag	Matches a specific tag value.
tag in	Conditionally compares tag-route against tag-set.
vpn-distinguisher is	Compares the VPN distinguisher against a specified value.

Table 2: Action Statements

Command	Description
abort (RPL)	Discards a route policy definition and returns to XR Config mode.
add	Adds an offset to an existing value.
apply	Executes a parameterized or an unparameterized policy from within another policy.
delete community	Deletes community values from a community list in a route.
delete extcommunity rt	Deletes extended community values from an extended community list in a route.
done	Accepts this route with no further processing
drop	Drops a route.
end-policy	Ends the definition of a route policy and exits route-policy configuration mode.
pass	Signifies that even though the route has not been modified, the user wants to continue executing in the policy block.
prepend as-path	Prepends the AS path with additional autonomous system numbers.
replace as-path	Replaces a sequence of AS numbers or private AS numbers in the AS path with the configured local AS.
set community	Sets the BGP community attribute.
set dampening	Configures BGP route dampening.
set extcommunity cost	Replaces or adds the extended communities for a cost on the route.

Command	Description
set extcommunity rt	Replaces or adds the extended communities for an RT on the route.
set ip-precedence	Sets the IP precedence to classify packets.
set isis-metric	Sets the IS-IS metric attribute value.
set label	Sets the BGP label attribute value.
set level	Configures the IS-IS level in which redistributed routes should be sent.
set local-preference	Specifies a preference value for the autonomous system path.
set med	Sets the MED value.
set metric-type (IS-IS)	Controls whether IS-IS treats the metric as an internal or external metric.
set metric-type (OSPF)	Controls whether OSPF treats the cost as a Type 1 or Type 2 metric.
set next-hop	Replaces the next-hop associated with a given route.
set origin	Changes the origin attribute.
set ospf-metric	Sets an OSPF protocol metric attribute value.
set qos-group (RPL)	Sets the QoS group to classify packets.
set rib-metric	Sets a RIB metric attribute value for a table policy.
set rip-metric	Sets RIP metric attributes.
set rip-tag	Sets route tag attribute.
set tag	Sets the tag attribute.
set traffic-index	Sets the traffic index attribute.
set weight	Sets the weight value for BGP routes.
suppress-route	Indicates that a given component of an aggregate should be suppressed, that is, not advertised.
unsuppress-route	Indicates that a given component of an aggregate should be unsuppressed.
set vpn-distinguisher	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/RP0/CPU0:router(config-rpl)# if as-path in as-path-set-1 then
RP/0/RP0/CPU0:router(config-rpl-if)# drop
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/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/RP0/CPU0:router(config-rpl)# if origin is igp then
RP/0/RP0/CPU0:router(config-rpl-if)# set med 42
RP/0/RP0/CPU0:router(config-rpl-if)# prepend as-path 73 5
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

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

```
RP/0/RP0/CPU0:router(config-rpl)# if med eq 200 then
RP/0/RP0/CPU0:router(config-rpl-if)# set community (12:34) additive
RP/0/RP0/CPU0:router(config-rpl-if)# else
RP/0/RP0/CPU0:router(config-rpl-else)# set community (12:56) additive
RP/0/RP0/CPU0:router(config-rpl-else)# endif
RP/0/RP0/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/RP0/CPU0:router(config-rpl)# if med eq 150 then
RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 10
RP/0/RP0/CPU0:router(config-rpl-if)# elseif med eq 200 then
RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 60
RP/0/RP0/CPU0:router(config-rpl-elseif)# elseif med eq 250 then
RP/0/RP0/CPU0:router(config-rpl-elseif)# set local-preference 110
RP/0/RP0/CPU0:router(config-rpl-elseif)# else
RP/0/RP0/CPU0:router(config-rpl-else)# set local-preference 0
RP/0/RP0/CPU0:router(config-rpl-else)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

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

```
RP/0/RP0/CPU0:router(config-rpl)# if community matches-any (12:34, 56:78) then
RP/0/RP0/CPU0:router(config-rpl-if)# if med eq 150 then
RP/0/RP0/CPU0:router(config-rpl-if)# drop
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/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 Descriminator (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

eq | **is** | **ge** | **le**	Equal to; exact match; greater than or equal to; less than or equal to.
number	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
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 7.0.12	This command was introduced.

Usage Guidelines

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

eq | **is** | **ge** | **le**	Equal to; exact match; greater than or equal to; less than or equal to.
number	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config-rpl)# if med eq 10 then
RP/0/RP0/CPU0:router(config-rpl-if)# set local-preference 100
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

next-hop-type is

To filter routes based on Cisco Multiprotocol Label Switching Traffic Engineering tunnels as the next-hop type, use the **next-hop-type is** command in route-policy configuration mode.

next-hop-type is mpls-te

Syntax Description

mpls-te	Filters BGP routes based on Cisco Multiprotocol Label Switching Traffic Engineering tunnels.

Command Default

No default behavior or values.

Command Modes

Route-policy configuration

Command History

Release	Modification
Release 25.1.1	This command was introduced.

Usage Guidelines

None.

Task ID

Task ID	Operations
route-policy	read, write

Examples

The example configuration allows only Cisco Multiprotocol Label Switching Traffic Engineering routes by filtering nexthops based on the next-hop-type.

```
Router(config)#route-policy ROUTE-RESOLUTION
Router(config-rpl)#if protocol is isis 100 and route-type is level-1 then
Router(config-rpl-if)#pass
Router(config-rpl-if)#elseif protocol is isis 100 and route-type is level-2 and next-hop-type
 is mpls-te then
Router(config-rpl-elseif)#pass
Router(config-rpl-elseif)#else
Router(config-rpl-else)#drop
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#commit
```

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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config-rpl)# pass
```

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

```
RP/0/RP0/CPU0:router(config-rpl)# if destination in permitted then
RP/0/RP0/CPU0:router(config-rpl-if)# pass
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

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 XR Config mode. To remove a named prefix set, use the **no** form of this command.

prefix-set *name*

Syntax Description

name	Name of a prefix set.

Command Default

None

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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 **g**reater than or **e**qual 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 **l**ess than or **e**qual 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 **eq**ual 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 mus 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/RP0/CPU0:router(config)# prefix-set legal-ipv4-prefix-examples
RP/0/RP0/CPU0:router(config-pfx)# 10.0.1.1,
RP/0/RP0/CPU0:router(config-pfx)# 10.0.2.0/24,
RP/0/RP0/CPU0:router(config-pfx)# 10.0.3.0/24 ge 28,
RP/0/RP0/CPU0:router(config-pfx)# 10.0.4.0/24 le 28,
RP/0/RP0/CPU0:router(config-pfx)# 10.0.5.0/24 ge 26 le 30,
RP/0/RP0/CPU0:router(config-pfx)# 10.0.6.0/24 eq 28
RP/0/RP0/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/RP0/CPU0:router(config)# prefix-set INVALID-PREFIX-EXAMPLES
RP/0/RP0/CPU0:router(config-pfx)# 10.1.1.1 ge 16,
RP/0/RP0/CPU0:router(config-pfx)# 10.1.2.1 le 16,
RP/0/RP0/CPU0:router(config-pfx)# 10.1.3.0/24 le 23,
RP/0/RP0/CPU0:router(config-pfx)# 10.1.4.0/24 ge 33,
RP/0/RP0/CPU0:router(config-pfx)# 10.1.5.0/25 ge 29 le 28
RP/0/RP0/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/RP0/CPU0:router(config)# prefix-set legal-ipv6-prefix-examples
RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:1::/64,
RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:2::/64,
RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:3::/64,
RP/0/RP0/CPU0:router(config-pfx)# 2001:0:0:4::/64
RP/0/RP0/CPU0:router(config-pfx)# end-set
```

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

```
RP/0/RP0/CPU0:router(config)# prefix-set legal-ipv4-prefix
RP/0/RP0/CPU0:router(config-pfx)# 10.1.1.1  0.255.0.255
RP/0/RP0/CPU0:router(config-pfx)# 10.2.2.2 0.0.0.0
RP/0/RP0/CPU0:router(config-pfx)# 10.3.3.3 255.255.255.254
RP/0/RP0/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.

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

prefix-set-name	Name of a prefix set.
inline-prefix-set	Inline prefix set. The inline prefix set must be enclosed in parentheses.
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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config-rpl)# if rib-has-route in (10.10.0.0/16 ge 16) then
RP/0/RP0/CPU0:router(config-rpl-if)# pass
RP/0/RP0/CPU0:router(config-rpl-if)# endif
RP/0/RP0/CPU0:router(config-rpl)#
```

route-policy (RPL)

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

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

Syntax Description

name	Name of a route policy.
parameter	(Optional) Parameter name. The parameter name must be preceded with a "$." The *parameters* must be enclosed in parenthesis "()".

Command Default

No default behavior or values

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config)# route-policy drop-everything
RP/0/RP0/CPU0:router(config-rpl)# drop
RP/0/RP0/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/RP0/CPU0:router(config)# route-policy drop-as-1234
RP/0/RP0/CPU0:router(config-rpl)# if as-path passes-through '1234' then
RP/0/RP0/CPU0:router(config-rpl-if)# apply check-communities
RP/0/RP0/CPU0:router(config-rpl-if)# else
RP/0/RP0/CPU0:router(config-rpl-else)# pass
RP/0/RP0/CPU0:router(config-rpl-else)# endif
```

```
RP/0/RP0/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.

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

community-set-name	Community set name.
inline-community-set	Inline community set. The inline community set must be enclosed in parentheses.
parameter	Parameter name. The parameter name must be preceded with a "$."
additive	(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 7.0.12	This command was introduced.

Usage Guidelines

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 **peeras** 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 **peeras** 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/RP0/CPU0:router(config-rpl)# set community (10:24)
RP/0/RP0/CPU0:router(config-rpl)# set community (10:24, $as:24, $as:$tag)
RP/0/RP0/CPU0:router(config-rpl)# set community (10:24, internet) additive
RP/0/RP0/CPU0:router(config-rpl)# set community (10:24, $as:24) additive
RP/0/RP0/CPU0:router(config-rpl)# set community (10:24, peeras:24) additive
```

set extcommunity transitive-bandwidth

To set and advertise the BGP DMZ external community attributes in a route with transitive-bandwidth extended community, use the **set extcommunity transitive-bandwidth** command in the route-policy configuration mode.

set extcommunity transitive-bandwidth { *parameter extcommunity-set-name* }

Syntax Description

extcommunity-set-name	External community set name.
parameter	Parameter name. You must precede the parameter name with a "$" when you want to use it as a variable.

Command Default

None.

Command Modes

Route-policy configuration.

Command History

Release	Modification
Release 25.1.1	The **transitive-bandwidth** type is introduced as an extended community in RPL.
Release 7.0.12	The **set extcommunity** command was introduced.

Task ID

Task ID	Operations
route-policy	read, write

Examples

The following is a configuration example using the **set extcommunity transitive-bandwidth** command, where you set and apply the transitive-bandwidth extended community for bandwidth advertisement in BGP:

```
Router#config
Router(config)#extcommunity-set transitive-bandwidth dmz-transitive
Router(config-ext)#1000:20000
Router(config-ext)#end-set
Router(config)#route-policy bgp_rpl_outbound
Router(config-rpl)#set extcommunity transitive-bandwidth dmz-transitive
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

number	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
parameter	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 6.0	This command was introduced.

Usage Guidelines

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 *Routing Configuration Guide for Cisco 8000 Series Routers* 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/RP0/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

number	Value assigned to a 32-bit unsigned integer. Range is 0 to 4294967295.
parameter	Parameter name. The parameter name must be preceded with a "$."
igp-cost	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.
max-reachable	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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config-rpl)# set med 156
RP/0/RP0/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/RP0/CPU0:router(config-rpl)# set med igp-cost
```

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 [destination-vrf] ipv6-address [destination-vrf]* | **discard** *parameter* | **peer-address** | **self** | **unchanged** }

Syntax Description

ipv4-address	Valid IPv4 address.
ipv6-address	Valid IPv6 address.
discard	Sets next-hop as Null0 interface.
destination-vrf	(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.
peer-address	Sets the next-hop to the IP address of the remote Border Gateway Protocol (BGP) peer.
parameter	Parameter name. The parameter name must be preceded with a "$."
self	Sets itself as the next-hop.
unchanged	Preserves the received next-hop without modification.

Command Default

No default behavior or values

Command Modes

Route-policy configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **set next-hop** command to replace the next-hop associated with a specific address.

The next hop destination is selected according to the address family. Example: for ipv4 address-family, the IPv4 address is used and for ipv6 address-family, the IPv6 address is used.

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.

The below address families support the selective setting of 'next-hop-self' via the RPL statement 'set next-hop self' starting in 4.2.1. Previous to this the setting of next-hop-self via an RPL was for all prefixes in the address family or none of the prefixes.

- IPv4 unicast
- IPv4 labeled-unicast
- IPv4 multicast
- IPv6 unicast
- IPv6 multicast

The **set next-hop discard** configuration is used in the neighbor inbound policy. When this config is applied to a path, the primary next-hop is still be associated with the actual path but the RIB is updated with next-hop set to Null0. Even if the primary received nexthop is unreachable, the Remotely Triggered Blackhole (RTBH) path will be considered reachable and will be a candidate in the bestpath selection process. The RTBH path is readvertised to other peers with either the received next-hop or nexthop-self based on normal BGP advertisement rules.

Use the **unchanged** keyword when you want the route-policy to leave the next-hop attribute exactly as it was received, without making any changes. This is useful in scenarios where retaining the original next-hop is required for proper routing or policy consistency.

Task ID

Task ID	Operations
route-policy	read, write

Examples

In the following example, the next-hop is set to a valid IPv4 address:

```
Router(config-rpl)# set next-hop 10.0.0.5
```

In this example, the next-hop is set to a parameter value $nexthop:

```
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:

```
Router(config-rpl)# set next-hop 10.0.0.5 destination-vrf
```

In the following example, the next-hop attribute is preserved as received by using the unchanged keyword in a route-policy:

```
Router(config-rpl)# set next-hop unchanged
```

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

igp	Sets the origin type to Interior Gateway Protocol (IGP).
incomplete	Sets the origin type to incomplete.
egp	Sets the origin type to Exterior Gateway Protocol (EGP).
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 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router(config-rpl)# set origin egp
```

show rpl

To display system-wide RPL configuration, use the **show rpl** command in XR EXEC mode.

show [**running-config**] **rpl** [**maximum** {**lines** *configuration-limit* | **policies** *policies-limit*} | **editor** {**emacs** | **nano** | **vim**}]

Syntax Description

running-config	(Optional) Displays configuration-limit argument.
maximum	(Optional) Displays the maximum number of lines of configuration and number of policies.
lines *configuration-limit*	(Optional) Displays the number of lines to which configuration is limited. Range is 1 to 131072. The *configuration-limit* argument is available if the **running-config** keyword is specified.
policies *policies-limit*	(Optional) Displays the limit on the number of policies. Range is 1 to 5000. The *configuration-limit* argument is available if the **running-config** keyword is specified.
editor	(Optional) Specifies the default RPL editor. This keyword is available if the **running-config** keyword is specified.
emacs	(Optional) Displays the default RPL editor to Micro Emacs.
nano	(Optional) Displays the default RPL editor to nano.
vim	(Optional) Displays the default RPL editor to Vim.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.11.1	The **Nano** and **Emacs** keyword was deprecated.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
route-policy	read, write

Examples

The following shows the output of the **show running-config rpl** command:

```
RP/0/RP0/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_exl) 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
!
```

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 XR EXEC mode.

show rpl active as-path-set [**detail**]

Syntax Description

detail	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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
```

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 XR EXEC mode.

show rpl active community-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for active community sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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
```

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 XR EXEC mode.

show rpl active extcommunity-set [**cost** | **rt** | **soo**] [**detail**]

Syntax Description

cost	(Optional) Displays all extended community cost sets.
rt	(Optional) Displays all extended community RT sets.
soo	(Optional) Displays all extended community SoO sets.
detail	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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
```

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 XR 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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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

The following prefix-sets are ACTIVE
------------------------------------
prefix_set_1
```

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

detail	(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 mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl active route-policy [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for active route policies.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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
```

show rpl as-path-set

To display the contents of AS path sets, use the **show rpl as-path-set** command in XR EXEC mode.

show rpl as-path-set [*name* | **states** | **brief**]

Syntax Description

name	(Optional) Name of the AS path set.
states	(Optional) Displays all unused, inactive, and active states.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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
```

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 XR EXEC mode.

show rpl as-path-set *name* **attachpoints**

Syntax Description

name	Name of an AS path set.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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.

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 XR EXEC mode.

show rpl as-path-set *name* **references** [**brief**]

Syntax Description

name	Name of the prefix set.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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.
policy status	Status of the policy.

show rpl community-set

To display the configuration of community sets, use the **show rpl community-set** command in XR EXEC mode.

show rpl community-set [*name* | **states** | **brief**]

Syntax Description

name	(Optional) Name of the community set.
states	(Optional) Shows all unused, inactive, and active states.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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

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/RP0/CPU0:router# show rpl community-set comm_set_ex2

community-set comm_set_ex2
  65501:1,
  65501:2,
  65501:3
end-set
```

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 XR EXEC mode.

show rpl community-set *name* **attachpoints**

Syntax Description

name	Name of a community set.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.

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 XR EXEC mode.

show rpl community-set *name* **references** [**brief**]

Syntax Description

name	Name of a community set.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.
usage	Type of usage for the policy.
policy status	Status of the policy.

show rpl extcommunity-set

To display the configuration of extended community sets, use the **show rpl extcommunity-set** command in XR EXEC mode.

show rpl extcommunity-set [*name* [**attachpoints** | **references**]] [**cost** | **rt** | **soo**] [*name*] [**brief**] [**states**]

Syntax Description

name	(Optional) Name of the community set.
attachpoints	(Optional) Displays all attach points for this community set.
references	(Optional) Displays all policies that use this community set.
cost	(Optional) Displays all extended community cost sets.
rt	(Optional) Displays all extended community RT sets.
soo	(Optional) Displays all extended community SoO sets.
brief	(Optional) Limits the display to a list of the names of all extended community sets without their configurations.
states	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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 rtsetl1
  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/RP0/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
!
```

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 XR 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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl inactive community-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for inactive community sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl inactive extcommunity-set [**detail**]

Syntax Description

detail	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl inactive prefix-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for inactive prefix sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 6.0	This command was introduced.

Usage Guidelines

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.

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 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/RP0/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
```

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 XR EXEC mode.

show rpl inactive rd-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for inactive RD sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR 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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **show rpl inactive route-policy** 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.

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/RP0/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
```

show rpl maximum

To display the maximum limits for lines of configuration and number of policies, use the **show rpl maximum** command in XR EXEC mode.

show rpl maximum [**lines** | **policies**]

Syntax Description

lines	(Optional) Displays the number of lines of configuration limit.
policies	(Optional) Displays the number of policies limit.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

show rpl maximum field descriptions 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.

Field	Description
Compiled policies size (kB)	Displays the current compiled total for policies in kilobytes.

show rpl policy-global references

To display policy-global definitions, use the **show rpl policy-global references** command in XR EXEC mode.

show rpl policy-global references [**brief**]

Syntax Description

brief	(Optional) Limits the display to a list of the policy names.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

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/RP0/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 XR EXEC mode.

show rpl prefix-set [*name* | **states** | **brief**]

Syntax Description

name	(Optional) Name of the prefix set.
states	(Optional) Shows all unused, inactive, and active states.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router# show rpl prefix-set pset1
!
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
```

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 XR EXEC mode.

show rpl prefix-set *name* **attachpoints**

Syntax Description

name	Name of a prefix set.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.

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 XR EXEC mode.

show rpl prefix-set *name* **references** [**brief**]

Syntax Description

name	Name of the prefix set.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.

show rpl rd-set

To display the configuration of route distinguisher (RD) sets, use the **show rpl rd-set** command in XR EXEC mode.

show rpl rd-set [*name* | **states** | **brief**]

Syntax Description

name	(Optional) Name of the RD set.
states	(Optional) Shows all unused, inactive, and active states.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router# show rpl rd-set rdset1

rd-set rdset1
  10:151,
  100.100.100.1:153,
  100.100.100.62/31:63
end-set
```

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 XR EXEC mode.

show rpl rd-set *name* **attachpoints**

Syntax Description

name	Name of an RD set.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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
  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/RP0/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.

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 XR EXEC mode.

show rpl rd-set *name* **references** [**brief**]

Syntax Description

name	Name of the RD set.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.

show rpl route-policy

To display the configuration of route policies, use the **show rpl route-policy** command in XR EXEC mode.

show rpl route-policy [*name* [**detail**] | **states** | **brief**]

Syntax Description

name	(Optional) Name of a route policy.
detail	(Optional) Displays the configuration of all policies and sets that a policy uses.
states	(Optional) Shows all unused, inactive, and active states.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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
```

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 XR EXEC mode.

show rpl route-policy *name* **attachpoints**

Syntax Description

name	Name of a policy.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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/RP0/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.

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 XR EXEC mode.

show rpl route-policy *name* **inline**

Syntax Description

name	Name of a policy.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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 XR EXEC mode.

show rpl route-policy *name* **references** [**brief**]

Syntax Description

name	Name of a prefix set.
brief	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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.

Field	Description
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.

show rpl route-policy uses

To display information about a specified named policy, use the **show rpl route-policy uses** command in XR EXEC mode.

show rpl route-policy *name* **uses** {**policies** | **sets** | **all**} [**direct**]

Syntax Description

name	Name of a policy.
policies	Generates a list of all policies that the named policy uses.
sets	Lists all named sets that are used by the policy.
all	Generates a list of both sets and policies that the named policy references.
direct	(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 **apply** statement are not listed.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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/RP0/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 XR EXEC mode.

show rpl unused as-path-set [**detail**]

Syntax Description

detail	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl unused community-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for unused community sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl unused extcommunity-set [**cost** | **detail** | **rt** | **soo**]

Syntax Description

cost	(Optional) Displays the unused extended-community cost objects.
rt	(Optional) Displays the unused extended community RT objects.
soo	(Optional) Displays the unused extended-community SoO objects.
detail	(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

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl unused prefix-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for unused prefix sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/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
```

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 XR EXEC mode.

show rpl unused rd-set [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for unused RD sets.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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.

Task ID

Task ID	Operations
route-policy	read

Examples

The **show rpl unused rd-set** command displays the following information:

```
RP/0/RP0/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.
```

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 XR EXEC mode.

show rpl unused route-policy [**detail**]

Syntax Description

detail	(Optional) Displays the content of the object and all referenced objects for unused route policies.

Command Default

No default behavior or values

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

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/RP0/CPU0:router# show run | begin prefix-set

Building configuration...
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
!
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/RP0/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
```

transmit (ePBR)

The command forwards packets through the normal IP routing table lookup if they match the associated class map criteria configured under a PBR class within a policy map.

transmit

Command Default

None.

Command Modes

Policy map configuration mode.

Command History

Release	Modification
Release 25.3.1	This command was introduced.

Task ID

Task ID	Operation
qos	read, write

```
Router(config)#policy-map type pbr INGRESS-SECURITY-POLICY
Router(config-pmap)#class type traffic MANAGEMENT-TRAFFIC
Router(config-pmap-c)#transmit
Router(config-pmap-c)#exit
```