

---

## Route Policy Command Reference

---

### Generic Commands

#### abort

**Syntax** `abort`

**Context** `config>router>policy-options`

This command is required to discard changes made to a route policy.

**Default** `none`

#### begin

**Syntax** `begin {exclusive}`

**Context** `config>router>policy-options`

**Description** This command is required in order to enter the mode to create or edit route policies.

**Default** `none`

**Parameters** `exclusive` — Specifies an exclusive lock on the policy configuration. Other CLI and SNMP users will be unable to edit the policy configuration until the lock is removed (via `commit`, `abort`, a timeout occurring, or a forced override).

#### commit

**Syntax** `commit`

**Context** `config>router>policy-options`

**Description** This command is required to save changes made to a route policy.

**Default** `none`

## Generic Commands

### description

<b>Syntax</b>	<b>description</b> <i>string</i> <b>no description</b>
<b>Context</b>	config>router>policy-options>policy-statement config>router>policy-options>policy-statement>entry
<b>Description</b>	This command creates a text description which is stored in the configuration file to help identify the content of the entity.  The <b>no</b> form of the command removes the string from the configuration.
<b>Default</b>	<b>none</b>
<b>Parameters</b>	<i>string</i> — The description character string. Allowed values are any string up to 80 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

---

## Route Policy Options

### as-path (policy options)

<b>Syntax</b>	<b>as-path</b> <i>name</i> <b>expression</b> <i>regular-expression</i> <b>no as-path</b> <i>name</i>
<b>Context</b>	config>router>policy-options
<b>Description</b>	This command creates a route policy AS path regular expression statement to use in route policy entries. The <b>no</b> form of the command deletes the AS path regular expression statement.
<b>Default</b>	No AS path regular expression statement is defined.
<b>Parameters</b>	<i>name</i> — The AS path regular expression name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. <i>reg-exp</i> — The AS path regular expression. Allowed values are any string up to 256 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, "@variable@". null — The AS path expressed as an empty regular expression string.

### as-path-group (policy options)

<b>Syntax</b>	<b>as-path-group</b> <i>name</i> <b>no as-path-group</b> <i>name</i>
<b>Context</b>	config>router>policy-options
<b>Description</b>	This command creates a route policy AS path regular expression statement to use in route policy entries. The <b>no</b> form of the command deletes the AS path regular expression statement.
<b>Default</b>	No AS path regular expression statement is defined.
<b>Parameters</b>	<i>name</i> — The AS path regular expression name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, "@variable@".

## Route Policy Options

### entry

<b>Syntax</b>	<b>entry</b> <i>entry-id</i> <b>expression</b> <i>reg-exp</i> <b>no entry</b>
<b>Context</b>	config>router>policy-options>as-path-group
<b>Description</b>	<p>This command creates the context to edit route policy entries within an autonomous system path group. Multiple entries can be created using unique entries. The router exits the filter when the first match is found and executes the action specified. For this reason, entries must be sequenced correctly from most to least explicit.</p> <p>An entry does not require matching criteria defined (in which case, everything matches) but must have at least define an action in order to be considered complete. Entries without an action are considered incomplete and will be rendered inactive.</p> <p>The <b>no</b> form of the command removes the specified entry from the autonomous system path group.</p>
<b>Default</b>	<b>none</b>
<b>Parameters</b>	<p><i>entry-id</i> — The entry ID expressed as a decimal integer. An <i>entry-id</i> uniquely identifies match criteria and the corresponding action. It is recommended that multiple entries be given <i>entry-ids</i> in staggered increments. This allows users to insert a new entry in an existing policy without requiring renumbering of all the existing entries.</p> <p><b>Values</b>     1 — 4294967295</p> <p><i>reg-exp</i> — The AS path group regular expression. Allowed values are any string up to 256 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.</p>

### community

<b>Syntax</b>	<b>community</b> <i>name</i> <b>members</b> <i>comm-id</i> [ <i>comm-id</i> ]...(up to 15 max) <b>community</b> <i>name</i> <b>expression</b> <i>expression</i> <b>no community</b> <i>name</i> [ <b>members</b> <i>comm-id</i> ]
<b>Context</b>	config>router>policy-options
<b>Description</b>	<p>This command creates a route policy community list to use in route policy entries.</p> <p>The <b>no</b> form of the command deletes the community list or the provided community ID.</p>
<b>Default</b>	<b>no community</b> — No community names or members are specified.
<b>Parameters</b>	<p><i>name</i> — The community list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.</p>

*comm-id* — The community ID. Note that up to 15 community ID strings can be specified up to a total maximum of 72 characters. A community ID can be specified in different forms.

**Values** 72 chars max

```
<2byte-asnumber:comm-val>|<reg-ex>|<ext-comm>|<well-known-comm>
ext-comm      <type>:  {<ip-address:comm-val>|
                       <reg-ex1&reg-ex2>|
                       <ip-address&reg-ex2>|
                       <2byte-asnumber:ext-comm-val>|
                       <4byte-asnumber:comm-val>|<as-number:val-in-mbps>}

ext:4300:<ovstate>
extL<value1>:<value2>
2byte-asnumber [0..65535]
comm-val       [0..65535]
reg-ex         [72 chars max]
type           target|origin
ip-address     a.b.c.d
ext-comm-val   [0..4294967295]
4byte-asnumber [0..4294967295]
reg-ex1        [63 chars max]
reg-ex2        [63 chars max]
well-known-comm null|no-export|no-export-subconfed| no-advertise
as-number      [0..65535]
val-in-mbps    [0..16777215]
ovstate        0, 1 or 2 (0 for valid), (1 for Not-Found), or (2 for Invalid)
value1         [0000..FFFF]
value2         [0..FFFFFFFFFFFFFF]
```

- *as-num:comm -value* — The *as-num* is the Autonomous System Number (ASN)

**Values** as-num: 1 — 65535  
comm-value: 0 — 65535

- type {target | origin}:as-num:comm-value — The keywords target or origin denote the community as an extended community of type route target or route origin respectively. The as-num and comm-value allow the same values as described above for regular community values, including regular expressions.
- *reg-ex1 reg-ex2* — A regular expression string. Allowed values are any string up to 63 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.
- *well-known-comm* — keywords **null**, **no-export**, **no-export-subconfed**, **no-advertise**.

**expression** *expression* — Specifies a logical expression containing terms and operators. It can contain sub-expressions enclosed in round brackets.

**Values** 900 chars max <expression> is one of the following:  
<expression> {AND|OR} <expression>  
[NOT] ( <expression> )  
[NOT] <comm-id>

For example:

**from community expression** "[community list A] OR ([community list B] AND [community list C])"

## Route Policy Options

### exclusive-lock-time

<b>Syntax</b>	<b>exclusive-lock-time</b> <i>seconds</i> <b>no exclusive-lock</b>
<b>Context</b>	config>router>policy-options
<b>Description</b>	This command specifies the inactivity timer for the exclusive lock time for policy editing. When a session is idle for greater than this time, the lock is removed and the configuration changes is aborted.
<b>Default</b>	300 seconds
<b>Parameters</b>	<i>seconds</i> — Specifies the duration the session with exclusive lock may be inactive.
	<b>Values</b> Values                      1 - 3600

### policy-options

<b>Syntax</b>	<b>[no] policy-options</b>
<b>Context</b>	config>router
<b>Description</b>	This command enables the context to configure route policies. Route policies are applied to the routing protocol.  The <b>no</b> form of the command deletes the route policy configuration.
<b>Default</b>	<b>none</b>

### triggered-policy

<b>Syntax</b>	<b>[no] triggered-policy</b>
<b>Context</b>	config>router
<b>Description</b>	This command triggers route policy re-evaluation.  By default, when a change is made to a policy in the <b>config router policy options</b> context and then committed, the change is effective immediately. There may be circumstances when the changes should or must be delayed; for example, if a policy change is implemented that would effect every BGP peer on a router, the consequences could be dramatic. It is more effective to control changes on a peer by peer basis.  If the <b>triggered-policy</b> command is enabled, and a given peer is established, and you want the peer to remain up, then, in order for a change to a route policy to take effect, a <b>clear</b> command with the <i>soft</i> or <i>soft-inbound</i> option must be used. In other words, when a <b>triggered-policy</b> is enabled, any routine policy change or policy assignment change within the protocol will not take effect until the protocol is reset or a clear command is issued to re-evaluate route policies; for example, <b>clear router bgp neighbor x.x.x.x soft</b> . This keeps the peer up and the change made to a route policy is applied only to that peer, or group of peers.
<b>Default</b>	Non-dynamic route policy is disabled.

---

## Route Policy Damping Commands

### damping

<b>Syntax</b>	<code>[no] damping <i>name</i></code>
<b>Context</b>	<code>config&gt;router&gt;policy-options</code>
<b>Description</b>	This command creates a context to configure a route damping profile to use in route policy entries. The <b>no</b> form of the command deletes the named route damping profile.
<b>Default</b>	<b>No damping profiles are defined.</b>
<b>Parameters</b>	<i>name</i> — The damping profile name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

### half-life

<b>Syntax</b>	<b>half-life <i>minutes</i></b> <b>no half-life</b>
<b>Context</b>	<code>config&gt;router&gt;policy-options&gt;damping</code>
<b>Description</b>	<p>This command configures the <b>half-life</b> parameter for the route damping profile.</p> <p>The half life value is the time, expressed in minutes, required for a route to remain stable in order for the Figure of Merit (FoM) value to be reduced by one half; for example, if the half life value is 6 (minutes) and the route remains stable for 6 minutes, then the new FoM value is 3 (minutes). After another 3 minutes pass and the route remains stable, the new FoM value is 1.5 (minutes).</p> <p>When the FoM value falls below the <b>reuse</b> threshold, the route is once again considered valid and can be reused or included in route advertisements.</p> <p>The <b>no</b> form of the command removes the half life parameter from the damping profile.</p>
<b>Default</b>	<b>No half life value is specified.</b> The half life value must be explicitly configured.
<b>Parameters</b>	<i>minutes</i> — The half life in minutes expressed as a decimal integer.
<b>Values</b>	1 — 45

## Route Policy Damping Commands

### max-suppress

**Syntax**    **max-suppress** *minutes*  
**no max-suppress**

**Context**    config>router>policy-options>damping

**Description**    This command configures the maximum suppression parameter for the route damping profile. This value indicates the maximum time, expressed in minutes, that a route can remain suppressed. The **no** form of the command removes the maximum suppression parameter from the damping profile.

**Default**    **No maximum suppression time is configured.**

**Parameters**    *minutes* — The maximum suppression time, in minutes, expressed as a decimal integer.

**Values**      1 — 720

### reuse

**Syntax**    **reuse** *integer*  
**no reuse**

**Context**    config>router>policy-options>damping

**Description**    This command configures the reuse parameter for the route damping profile. When the Figure of Merit (FoM) value falls below the **reuse** threshold, the route is once again considered valid and can be reused or included in route advertisements. The **no** form of the command removes the reuse parameter from the damping profile.

**Default**    **No reuse parameter is configured.**

**Parameters**    *integer* — The reuse value expressed as a decimal integer.

**Values**      1 — 20000



## suppress

**Syntax** **suppress** *integer*  
**no suppress**

**Context** config>router>policy-options>damping

**Description** This command configures the suppression parameter for the route policy damping profile.  
A route is suppressed when it has flapped frequently enough to increase the Figure of Merit (FoM) value to exceed the **suppress** threshold limit. When the **FoM** value exceeds the **suppress** threshold limit, the route is removed from the route table or inclusion in advertisements.

The **no** form of the command removes the suppress parameter from the damping profile.

**Default** No suppress parameter is configured.

**Parameters** *integer* — The suppress value expressed as a decimal integer.

**Values** 1 — 20000

---

## Route Policy Prefix Commands

### prefix-list

<b>Syntax</b>	<b>[no] prefix-list</b> <i>name</i>
<b>Context</b>	config>router>policy-options
<b>Description</b>	This command creates a context to configure a prefix list to use in route policy entries. The <b>no</b> form of the command deletes the named prefix list.
<b>Default</b>	<b>none</b>
<b>Parameters</b>	<i>name</i> — The prefix list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

An empty prefix list can be configured for pre-provisioning. This empty prefix list will not find a match when referred to by a policy. When removing member prefixes from a prefix list, the prefix list will not be automatically removed when the last member is removed. If required, an empty prefix list must be explicitly removed using the **no** form of the command.

### prefix

<b>Syntax</b>	<b>[no] prefix</b> <i>ip-prefix</i> / <i>prefix-length</i> { [ <b>exact</b>   <b>longer</b>   <b>through</b> <i>length</i> ] [ <b>prefix-length-range</b> <i>length1-length2</i> ] }														
	<b>no prefix</b> [ <i>ipv-prefix/prefix-length</i> ] [ <b>exact</b>   <b>longer</b>   <b>through</b> <i>length</i>   <b>prefix-length-range</b> <i>length1-length2</i> ]														
<b>Context</b>	config>router>policy-options>prefix-list														
<b>Description</b>	This command creates a prefix entry in the route policy prefix list. The <b>no</b> form of the command deletes the prefix entry from the prefix list.														
<b>Parameters</b>	<i>ip-prefix</i> — The IP prefix for prefix list entry in dotted decimal notation.														
<b>Values</b>	<table> <tr> <td>ipv4-prefix:</td> <td>a.b.c.d (host bits must be 0)</td> </tr> <tr> <td>ipv4-prefix-length:</td> <td>0 — 32</td> </tr> <tr> <td>ipv6-prefix:</td> <td>x:x:x:x:x:x:x (eight 16-bit pieces)</td> </tr> <tr> <td></td> <td>x:x:x:x:x:d.d.d.d</td> </tr> <tr> <td></td> <td>x: [0 — FFFF]H</td> </tr> <tr> <td></td> <td>d: [0 — 255]D</td> </tr> <tr> <td>ipv6-prefix-length:</td> <td>0 — 128</td> </tr> </table>	ipv4-prefix:	a.b.c.d (host bits must be 0)	ipv4-prefix-length:	0 — 32	ipv6-prefix:	x:x:x:x:x:x:x (eight 16-bit pieces)		x:x:x:x:x:d.d.d.d		x: [0 — FFFF]H		d: [0 — 255]D	ipv6-prefix-length:	0 — 128
ipv4-prefix:	a.b.c.d (host bits must be 0)														
ipv4-prefix-length:	0 — 32														
ipv6-prefix:	x:x:x:x:x:x:x (eight 16-bit pieces)														
	x:x:x:x:x:d.d.d.d														
	x: [0 — FFFF]H														
	d: [0 — 255]D														
ipv6-prefix-length:	0 — 128														
	<b>exact</b> — Specifies the prefix list entry only matches the route with the specified <i>ip-prefix</i> and prefix <i>mask</i> (length) values.														

**longer** — Specifies the prefix list entry matches any route that matches the specified *ip-prefix* and prefix *mask* length values equal to or greater than the specified mask.

**through** *length* — Specifies the prefix list entry matches any route that matches the specified ip-prefix and has a prefix length between the specified *length* values inclusive.

**Values** 0 — 32

**prefix-length-range** *length1 - length2* — Specifies a route must match the most significant bits and have a prefix length with the given range. The range is inclusive of start and end values.

**Values** 0 — 32, *length2* > *length1*

---

## Route Policy Entry Match Commands

### entry

**Syntax**    **entry** *entry-id*  
              **no entry**

**Context**    config>router>policy-options>policy-statement

**Description**    This command creates the context to edit route policy entries within the route policy statement.

Multiple entries can be created using unique entries. The router exits the filter when the first match is found and executes the action specified. For this reason, entries must be sequenced correctly from most to least explicit.

An entry does not require matching criteria defined (in which case, everything matches) but must have at least define an action in order to be considered complete. Entries without an action are considered incomplete and will be rendered inactive.

The **no** form of the command removes the specified entry from the route policy statement.

**Default**    **none**

**Parameters**    *entry-id* — The entry ID expressed as a decimal integer. An *entry-id* uniquely identifies match criteria and the corresponding action. It is recommended that multiple entries be given *entry-ids* in staggered increments. This allows users to insert a new entry in an existing policy without requiring renumbering of all the existing entries.

**Values**        1 — 4294967295

### area

**Syntax**    **area** *area-id*  
              **no area**

**Context**    config>router>policy-options>policy-statement>entry>from

**Description**    This command configures an OSPF area as a route policy match criterion.

This match criterion is only used in export policies.

All OSPF routes (internal and external) are matched using this criterion if the best path for the route is by the specified area.

The **no** form of the command removes the OSPF area match criterion.

**Default**    **none**

**Parameters**    *area-id* — The OSPF area ID expressed in dotted decimal notation or as a 32-bit decimal integer.

**Values**        0.0.0.0 — 255.255.255.255 (dotted decimal), 0 — 4294967295 (decimal)

## as-path

**Syntax** **as-path** *name*  
**no as-path**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command configures an AS path regular expression statement as a match criterion for the route policy entry.

If no AS path criterion is specified, any AS path is considered to match.

AS path regular expression statements are configured at the global route policy level (**config>router>policy-options>as-path** *name*).

The **no** form of the command removes the AS path regular expression statement as a match criterion.

**Default** **no as-path** — Matches any AS path.

**Parameters** *name* — The AS path regular expression name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

## as-path-group

**Syntax** **as-path-group** *name*  
**no as-path-group** *name*

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command creates a route policy AS path regular expression statement to use in route policy entries.

The **no** form of the command deletes the AS path regular expression statement.

**Default** No AS path regular expression statement is defined.

**Parameters** *name* — The AS path regular expression name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

## Route Policy Entry Match Commands

### community

<b>Syntax</b>	<b>community</b> <i>name</i> <b>no community</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry>from
<b>Description</b>	This command configures a community list as a match criterion for the route policy entry. If no community list is specified, any community is considered a match. The <b>no</b> form of the command removes the community list match criterion.
<b>Default</b>	<b>no community</b> — Matches any community.
<b>Parameters</b>	<i>name</i> — The community list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@," "start@variable@end"," @variable@end", or "start@variable@".  The <i>name</i> specified must already be defined.

### from

<b>Syntax</b>	<b>[no] from</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry
<b>Description</b>	This command creates the context to configure policy match criteria based on a route's source or the protocol from which the route is received. If no condition is specified, all route sources are considered to match. The <b>no</b> form of the command deletes the source match criteria for the route policy statement entry.

### external

<b>Syntax</b>	<b>[no] external</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry>from
<b>Description</b>	This command specifies the external route matching criteria for the entry.
<b>Default</b>	<b>no external</b>

## family

**Syntax** **family** [ipv4] [ipv6] [mcast-ipv4] [mcast-ipv6] [vpn-ipv4] [vpn-ipv6] [l2-vpn] [mvpn-ipv4] [mvpn-ipv6] [mdt-safi] [flow-ipv4] [flow-ipv6] [route-target] [mcast-vpn-ipv4] [evpn]  
**no family**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command specifies address families as matching conditions.

**Parameters**

- evpn** — Specifies Ethernet VPN related information.
- ipv4** — Specifies IPv4 routing information.
- ipv6** — Specifies IPv6 routing information.
- mcast-ipv4** — Specifies multicast IPv4 routing information.
- mcast-ipv6** — Specifies multicast IPv6 routing information.
- vpn-ipv4** — Specifies IPv4 VPN routing information.
- l2-vpn** — Exchanges Layer 2 VPN information.
- mvpn-ipv4** — Exchanges Multicast VPN related information
- mvpn-ipv6** — Exchanges Multicast VPN related information
- mdt-safi** — Exchange Multicast VPN (MDT-SAFI) related information
- flow-ipv4** — Exchanges IPv4 flowspec routes belonging to AFI 1 and SAFI 133
- flow-ipv6** — Exchange Ipv6 flowspec routes belonging to AFI 2 and SAFI 133
- route-target** — Specifies to use route targets to be advertised to the peers if ORF is enabled for this peer group
- mcast-vpn-ipv4** — — Exchanges Multicast Routes in VPN using SAFI 129.

## | flow-spec-dest

**Syntax** **flow-spec-dest** *prefix-list-name*  
**no flow-spec-dest**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command is used to match BGP flow-spec routes on the basis of the destination IP prefix in the flow specification. An IPv4 flow-spec route is matched by this command if its NLRI contains a type 1 subcomponent encoding a prefix and prefix-length that is covered by an entry in the referenced prefix-list. An IPv6 flow-spec route is matched by this command if its NLRI contains a type 1 component encoding prefix-off-set=0 and a prefix & prefix-length that is covered by an entry in the referenced prefix-list.

The **flow-spec-dest** command has no effect when the policy is not applied as a BGP import or export policy.

**Default** **no flow-spec-dest**

**Parameters** *prefix-list-name* — The name of a prefix-list containing IPv4 and/or IPv6 prefix entries [32 characters max].

## Route Policy Entry Match Commands

### flow-spec-source

**Syntax** **flow-spec-source** *prefix-list-name*  
**no flow-spec-source**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command is used to match BGP flow-spec routes on the basis of the source IP prefix in the flow specification. An IPv4 flow-spec route is matched by this command if its NLRI contains a type 2 subcomponent encoding a prefix and prefix-length that is covered by an entry in the referenced prefix-list. An IPv6 flow-spec route is matched by this command if its NLRI contains a type 2 component encoding prefix-offset=0 and a prefix & prefix-length that is covered by an entry in the referenced prefix-list.

The **flow-spec-source** command has no effect when the policy is not applied as a BGP import or export policy.

**Default** **no flow-spec-source**

**Parameters** *prefix-list-name* — The name of a prefix-list containing IPv4 and/or IPv6 prefix entries [32 characters max].

### group-address

**Syntax** **group-address** *prefix-list-name*  
**no group-address**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command specifies the multicast group-address prefix list containing multicast group-addresses that are imbedded in the join or prune packet as a filter criterion. The prefix list must be configured prior to entering this command. Prefix lists are configured in the **config>router>policy-options>prefix-list** context.

The **no** form of the command removes the criterion from the configuration.

**Default** **no group-address**

**Parameters** *prefix-list-name* — The prefix-list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

The *prefix-list-name* is defined in the **config>router>policy-options>prefix-list** context.

### host-ip

**Syntax** **host-ip** *prefix-list-name*

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command specifies a prefix list host IP address as a match criterion for the route policy-statement entry.

**Default** **no host-ip**



**Parameters** *prefix-list-name* — The prefix-list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

The *prefix-list-name* is defined in the **config>router>policy-options>prefix-list** context.

## interface

**Syntax** **interface** *interface-name*  
**no interface**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command specifies the router interface, specified either by name or address, as a filter criterion. The **no** form of the command removes the criterion from the configuration.

**Default** **no interface**

**Parameters** *ip-int-name* — Specify the name of the interface as a match criterion for this entry. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

## level

**Syntax** **level** {1 | 2}  
**no level**

**Context** config>router>policy-options>policy-statement>entry>from  
config>router>policy-options>policy-statement>entry>to

**Description** This command specifies the ISIS route level as a match criterion for the entry.

**Default** **no level**

**Parameters** 1 | 2 — Matches the IS-IS route learned from level 1 or level 2.

## mvpn-type

**Syntax** **mvpn-type** {1 | 2 | 3 | 4 | 5 | 6 | 7 }  
**no mvpn-type**

**Context** config>router>policy-options>polic-statement>entry>from

**Description** This command allows match on ng-MVPN BGP route type when the policy is used for VRF-import/VRF-export/BGP global export policy. The policy will only be applied to multicast routes.

The **no** form of the command disables **mvpn-type** in the policy evaluation.

**Default** **no mvpn-type**

## Route Policy Entry Match Commands

**Parameters** 1 | 2 | 3 | 4 | 5 | 6 | 7 — BGP MVPN route-type as per RFC6514.

### neighbor

**Syntax** **neighbor** {*ip-address* | **prefix-list** *name*}  
**no neighbor**

**Context** config>router>policy-options>policy-statement>entry>to  
config>router>policy-options>policy-statement>entry>from

**Description** This command specifies the neighbor address as found in the source address of the actual join and prune message as a filter criterion. If no neighbor is specified, any neighbor is considered a match.

The **no** form of the of the command removes the neighbor IP match criterion from the configuration.

**Default** **no neighbor** — Matches any neighbor.

**Parameters** *ip-addr* — The neighbor IP address in dotted decimal notation.

<b>Values</b>	ipv4-address:	a.b.c.d
	ipv6-address:	x:x:x:x:x:x[-interface] x:x:x:x:x:d.d.d.d[-interface] x: [0 — FFFF]H d: [0 — 255]D interface (32 chars max, mandatory for link local addresses)

**prefix-list** *name* — The prefix-list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

The *name* specified must already be defined.

### origin

**Syntax** **origin** {**igp** | **egp** | **incomplete** | **any** | **aaa** | **dhcp** | **ludb**}  
**no origin**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command will configure a match criteria for the origin attribute. Originally, the origin attribute was applicable only to BGP as a mandatory well-known BGP attribute.

The functionality of the origin attribute has expanded to subscriber-management routes (/32 IPv4 host and IPv6 PD wan-host routes). Each subscriber-management route will internally (local to the node) by default carry the origin attribute with one of the three new values (aaa, dhcp and ludb). The value of the attribute will depend on the origin of the subscriber-management route. The aaa, dhcp or ludb values will never be carried in BGP updates as part of the BGP origin attribute or be otherwise visible within the BGP process.

This introduction of the three new values for the origin attribute in the subscriber-management routes will allow customized advertisement of the subscriber-management routes via routing policy.

**Default** **no origin** — Matches any BGP origin attribute

**Parameters** **igp** — Configures matching path information originating within the local AS.

**egp** — Configures matching path information originating in another AS.

**incomplete** — Configures matching path information learned by another method.

**any** — Specifies to ignore this criteria.

**aaa** — IPv4

subscriber-management /32 host routes that are originated via Radius framed-ip-address VSA other than 255.255.255.254. The 255.255.255.254 returned by the Radius indicates that the BNG (NAS) should assign an IP address from its own pool.

IPv6

subscriber-management routes that are originated through framed-ipv6-prefix (SLAAC), delegated-ipv6-prefix (IA\_PD) or alc-ipv6-address (IA\_NA) Radius attributes . This is valid for IPoE and PPPoE type host.

**dhcp** — IPv4

subscriber-management /32 host routes that are originated via DHCP server (local or remote) and also Radius framed-ip-address=255.255.255.254 (RFC 2865).

IPv6

subscriber-management routes that are assigned via local DHCPv6 server pools whose name is obtained through Alc-Delegated-IPv6-Pool (PD pool) and Framed-IPv6-Pool (NA pool) Radius attributes. This is valid for IPoE and PPPoE type hosts.

In addition, for IPoEv6 only, the pool name can be also obtained via ipv6-delegated-prefix-pool (PD pool) and ipv6-wan-address-pool (NA pool) from LUDB.

**ludb** — IPv4

subscriber-management /32 host routes that are originated via LUDB. This should also cover Radius fallback category (Radius falls back to system-defaults or to LUDB).

IPv6

subscriber-management routes obtained from LUDB via ipv6-address (IA\_NA) or ipv6-prefix (IA\_PD). This is supported only for IPoE.

## origin-validation-state

**Syntax** **origin-validation-state** *state*  
**no origin-validation-state**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command is used to match BGP routes on the basis of origin validation state:

- Valid (0)
- Not-Found (1)

## Route Policy Entry Match Commands

- Invalid (2)

**Default** **no origin-validation-state**

**Parameters** **valid** — Marks the route as having an origin validation state of valid.  
**notFound** — Marks the route as having an origin validation state of Not Found.  
**invalid** — Marks the route as having an origin validation state of invalid.

## policy-statement

**Syntax** **[no] policy-statement** *name*

**Context** config>router>policy-options

**Description** This command creates the context to configure a route policy statement.  
Route policy statements control the flow of routing information to and from a specific protocol, set of protocols, or to a specific BGP neighbor.  
The **policy-statement** is a logical grouping of match and action criteria. A single **policy-statement** can affect routing in one or more protocols and/or one or more protocols peers/neighbors. A single **policy-statement** can also affect both the import and export of routing information.  
The **no** form of the command deletes the policy statement.

**Default** **no policy-statement** — No route policy statements are defined.

**Parameters** *name* — The route policy statement name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

## policy-variables

**Syntax** **policy-variables name** *name-string* **value** *value-string*  
**policy-variables name** *name-string* **address** *ip-address*  
**policy-variables name** *name-string* **number** *value-number*  
**no policy-variables name** *name-string*

**Context** config>router>policy-options>policy-statement>from

**Description** Routing policy variable allows operators a powerful and flexible configuration approach to routing policies for policies are often reused across BGP peers of a common type (transit; peer; customer; etc). Using policy variables allows an operator to have a single policy that is consistent across all peers of a type, while retaining the flexibility to reference different policy functions (prefixes, prefix-lists, community lists, etc) with unique names if required, by defining variable names and the variable value.

Depending on the parameter referenced, the correct type should be specified as follows:

- value-string: **as-path**, **as-path-group**, **community**, **prefix-list**, **damping**
- ip-address: **next-hop**

• value-number: **aigp-metric, as-path-prepend, local-preference, metric, origin, origin-validation, preference, tag, type**

The **no** form of the command removes the **policy-variables** statement.

### Parameters

*name-string* — The name of the policy variable, with the variable delimited by at-signs (@) at the beginning and the end of the name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

*value-string* — The value of the policy variable. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes.

*value-number* — The numerical value of the policy variable.

**Values**      0 — 4294967295

*ip-address* — The IP address of the policy variable.

**Values**      ipv4-address - a.b.c.d  
                   ipv6-address - x:x:x:x:x:x:x (eight 16-bit pieces)  
                   x:x:x:x:x:d.d.d.d  
                   x - [0..FFFF]H  
                   d - [0..255]D

## prefix-list

**Syntax**      **prefix-list** *name* [*name...*up to 5 max]  
**no prefix-list**

**Context**      config>router>policy-options>policy-statement>entry>from  
                   config>router>policy-options>policy-statement>entry>to

**Description**      This command configures a prefix list as a match criterion for a route policy statement entry.

If no prefix list is specified, any network prefix is considered a match.

An empty prefix list will evaluate as if 'no match' was found.

The prefix lists specify the network prefix (this includes the prefix and length) a specific policy entry applies.

A maximum of five prefix names can be specified.

The **no** form of the command removes the prefix list match criterion.

**Default**      **no prefix-list** — Matches any network prefix.

**Parameters**      *name* — The prefix list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

## Route Policy Entry Match Commands

### protocol

<b>Syntax</b>	<b>protocol</b> { <i>protocol</i> } [ <b>all</b>   <b>instance</b> <i>instance</i> ] <b>no protocol</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry>from
<b>Description</b>	This command configures a routing protocol as a match criterion for a route policy statement entry. This command is used for both import and export policies depending how it is used. If no protocol criterion is specified, any protocol is considered a match. The <b>no</b> form of the command removes the protocol match criterion.
<b>Default</b>	<b>no protocol</b> — Matches any protocol.
<b>Parameters</b>	<b>protocol</b> <i>protocol</i> — The protocol name to match on. <b>Values</b> direct, static, bgp, isis, ospf, rip, aggregate, bgp-vpn, igmp, pim, ospf3, ldp, sub-mgmt, mld, managed, vpn-leak, tms, nat, periodic, ipsec <b>instance</b> <i>instance</i> — The OSPF or IS-IS instance. <b>Values</b> 1 — 31 <b>all</b> — OSPF- or ISIS-only keyword.

### protocol

<b>Syntax</b>	<b>protocol</b> { <i>protocol</i> } [ <b>all</b>   <b>instance</b> <i>instance</i> ] <b>no protocol</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry>to
<b>Description</b>	This command configures a routing protocol as a match criterion for a route policy statement entry. This command is used for both import and export policies depending how it is used. If no protocol criterion is specified, any protocol is considered a match. The <b>no</b> form of the command removes the protocol match criterion.
<b>Default</b>	<b>no protocol</b> — Matches any protocol.
<b>Parameters</b>	<b>protocol</b> <i>protocol</i> — The protocol name to match on. <b>Values</b> bgp, isis, ospf, rip, bgp-vpn, ospf3, vpn-leak, ldp <b>instance</b> <i>instance</i> — The OSPF or IS-IS instance. <b>Values</b> 1 — 31 <b>all</b> — OSPF- or ISIS-only keyword.

## source-address

**Syntax** **source-address** *ip-address*  
**source-address prefix-list** *prefix-list-name*  
**no source-address**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command specifies the source address that is embedded in the join or prune packet as a filter criterion. The **no** form of the command removes the criterion from the configuration.

**Default** none

**Description** This command specifies a multicast data source address as a match criterion for this entry.

**Parameters** *ip-address* — The IP prefix for the IP match criterion in dotted decimal notation.

<b>Values</b>	ipv4-address	a.b.c.d
	ipv6-address	x:x:x:x:x:x:x (eight 16-bit pieces)
		x:x:x:x:x:d.d.d.d
		x - [0..FFFF]H
		d - [0..255]D

*prefix-list-name* — Specifies the prefix list name up to 32 characters in length.

## state

**Syntax** **state** *state*  
**no state**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command will configure a match criteria on the state attribute. The state attribute carries the state of an SRRP instance and it can be applied to:

- subscriber-interface routes
- subscriber-management routes (/32 IPv4 and IPv6 PD wan-host)
- managed-routes (applicable only to IPv4).

Based on the state attribute of the route we can manipulate the route advertisement into the network.

We can enable or disable (in case there is no SRRP running) tracking of SRRP state by routes.

This is done on a per subscriber-interface route basis, where a subscriber-interface route is tracking a single SRRP instance state (SRRP instance might be in a Fate Sharing Group).

For subscriber-management and managed-routes, tracking is enabled per group interface under which SRRP is enabled.

**Default** none

**Description** This command specifies a multicast data source address as a match criterion for this entry.

## Route Policy Entry Match Commands

- Parameters**
- srrp-master** — Track routes with the state attribute carrying srrp-master state.
  - srrp-non-master** — Track routes with the state attribute carrying srrp-non-master state.
  - ipsec-master-with-peer** — Track routes with the state attribute carrying ipsec-master-with-peer state.
  - ipsec-non-master** — Track routes with the state attribute carrying ipsec-non-master state.
  - ipsec-master-without-peer** — Track routes with the state attribute carrying ipsec-master-without-peer state.

### tag

**Syntax** **tag** *tag*  
**no tag**

**Context** config>router>policy-options>policy-statement>entry>from

**Description** This command matches the tag value in static or IGP routes. A decimal or hexadecimal value of 4 octets can be entered. For IS-IS, OSPF, and static routes, all four octets can be used. For RIP and RIPng, only the two most significant octets are used if more than two octets are configured.

The **no** form of the command removes the tag field match criterion.

**Default** **no tag** — Matches any tag value.

**Parameters** *tag* — Matches the configured tag value.

**Values** Accepts decimal or hexadecimal formats:  
IS-IS, OSPF and static routes: 0x0 – 0xFFFFFFFF or 1 – 4294967295  
RIP and RIPng: 0x0 – 0xFFFF or 1 – 65535

### to

**Syntax** [**no**] **to**

**Context** config>router>policy-options>policy-statement>entry

**Description** This command creates the context to configure export policy match criteria based on a route's destination or the protocol into which the route is being advertised.

If no condition is specified, all route destinations are considered to match.

The **to** command context only applies to export policies. If it is used for an import policy, match criteria is ignored.

The **no** form of the command deletes export match criteria for the route policy statement entry.



## type

**Syntax**    **type {1 | 2}**  
              **no type**

**Context**    config>router>policy-options>policy-statement>entry>from

**Description**    This command configures an OSPF type metric as a match criterion in the route policy statement entry. If no type is specified, any OSPF type is considered a match. The **no** form of the command removes the OSPF type match criterion.

**Parameters**    **1** — Matches OSPF routes with type 1 LSAs.  
                  **2** — Matches OSPF routes with type 2 LSAs.

---

## Route Policy Action Commands

### action

<b>Syntax</b>	<b>action</b> { <b>accept</b>   <b>next-entry</b>   <b>next-policy</b>   <b>reject</b> } <b>no action</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry
<b>Description</b>	<p>This command creates the context to configure actions to take for routes matching a route policy statement entry.</p> <p>This command is required and must be entered for the entry to be active.</p> <p>Any route policy entry without the <b>action</b> command will be considered incomplete and will be inactive.</p> <p>The <b>no</b> form of the command deletes the action context from the entry.</p>
<b>Default</b>	<b>no action</b> — No action is defined.
<b>Parameters</b>	<p><b>accept</b> — Specifies routes matching the entry match criteria will be accepted and propagated.</p> <p><b>next-entry</b> — Specifies that the actions specified would be made to the route attributes and then policy evaluation would continue with next policy entry (if any others are specified).</p> <p><b>next-policy</b> — Specifies that the actions specified would be made to the route attributes and then policy evaluation would continue with next route policy (if any others are specified).</p> <p><b>reject</b> — Specifies routes matching the entry match criteria would be rejected.</p>

### add-paths-send-limit

<b>Syntax</b>	<b>add-paths-send-limit</b> <i>send-limit</i> <b>no add-paths-send-limit</b>
<b>Context</b>	config>router>policy-options>policy-statement>entry config>router>policy-options>policy-statement>default-action
<b>Description</b>	<p>This command sets the Add-Paths send-limit to a specific value for all routes matched by the policy entry or default action. Add-Paths allows a BGP router to send multiple paths for the same NLRI/prefix to a peer advertising the Add-Paths receive capability. The send-limit dictates the maximum number of paths that can be advertised.</p> <p>The default send-limit is controlled by the instance, group or neighbor level configuration and applies to all prefixes in a particular address family. Using route policies allows the default send-limit to be overridden to use a larger or smaller maximum value on a per-prefix basis. For example if, for most prefixes advertised to a peer, at most 1 path should be advertised but for a few exceptional prefixes up to 4 paths should be advertised then the neighbor-level send-limit can be set to a value of 1 and the add-paths-send-limit in the policy entry that matches the exceptional routes can be set to a value of 4.</p>

**Default** no default

**Parameters** *send-limit* — Specify the maximum number of paths to advertise for matched routes to an Add-Paths peer.

**Values** 1—16

## advertise-label

**Syntax** **advertise-label per-prefix**  
**no advertise-label**

**Context** config>router>policy-options>policy-statement>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** When this command is configured as a default-action or entry-specific action of a VRF export policy, every qualifying matched route is advertised with a per-prefix label in the resulting VPN-IP routes. Examples of non-qualifying routes that are not affected by this command are local interface routes and BGP-VPN routes. Essentially this command overrides, for specific routes, the configured label-mode of the exporting VPRN service.

**Default** no advertise-label

**Parameters** **per-prefix** — Mandatory parameter that forces a per-prefix label allocation policy for matched routes.

## aigp-metric

**Syntax** **aigp-metric metric**  
**aigp-metric igp**  
**no aigp-metric**

**Context** config>router>policy-options>policy-statement>entry>action  
config>router>policy-options>policy-statement>default-action

**Description** The effect of this command on a route matched and accepted by a route policy entry depends on how the policy is applied (BGP import policy vs. BGP export policy), the type of route and the specific form of the command.

In a BGP import policy this command is used to:

- Associate an AIGP metric with an IBGP route received with an empty AS path and no AIGP attribute.
- Associate an AIGP metric with an EBGP route received without an AIGP attribute that has an AS path containing only AS numbers belonging to the local AIGP administrative domain.
- Modify the received AIGP metric value prior to BGP path selection

In a BGP export policy this command is used to:

- Add the AIGP attribute and set the AIGP metric value in a BGP route originated by exporting a direct, static or IGP route from the routing table
- Remove the AIGP attribute from a route advertisement to a particular peer

## Route Policy Action Commands

- Modify the AIGP metric value in a route advertisement to a particular peer

**Default** no aigp-metric

**Parameters** *metric* — Administratively defined metric.

**Values** 0 — 4294967295

**Default** none

*name*—The AIGP metric parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, "@variable@".

**igp** — Sets the AIGP metric to the IGP metric.

## as-path

**Syntax** **as-path** {**add** | **replace**} *name*  
**no as-path**

**Context** config>router>policy-options>policy-statement>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command assigns a BGP AS path list to routes matching the route policy statement entry. If no AS path list is specified, the AS path attribute is not changed. The **no** form of the command disables the AS path list editing action from the route policy entry.

**Default** **no as-path** — The AS path attribute is not changed.

**Parameters** **add** — Specifies that the AS path list is to be prepended to an existing AS list.

**replace** — Specifies AS path list replaces any existing as path attribute.

*name*—The AS path list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

The *name* specified must already be defined.

## as-path-prepend

**Syntax** **as-path-prepend** *as-num* [*repeat*]  
**no as-path-prepend**

**Context** config>router>policy-options>policy-statement>default-action  
 config>router>policy-options>policy-statement>entry>action

**Description** The command prepends a BGP AS number once or numerous times to the AS path attribute of routes matching the route policy statement entry.

If an AS number is not configured, the AS path is not changed.

If the optional *number* is specified, then the AS number is prepended as many times as indicated by the number.

The **no** form of the command disables the AS path prepend action from the route policy entry.

**Default** **no as-path-prepend** — no AS number prepending configured.

**Parameters** *as-num* — The AS number to prepend expressed as a decimal integer.

**Values** 1 — 4294967295  
 name—The AS path parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

*repeat* — The number of times to prepend the specified AS number expressed as a decimal integer.

**Values** 1 — 50

## bgp-leak

**Syntax** **bgp-leak**  
**no bgp-leak**

**Context** config>router>policy-options>policy-statement>default-action  
 config>router>policy-options>policy-statement>entry>action

**Description** This command causes qualifying matched BGP routes to be marked as leakable, meaning they are candidates to be leaked into other routing instances (copied with their complete set of path attributes). A BGP route is a qualifying route if the NLRI has an IPv4 or IPv6 prefix without a label. Note that a leakable BGP route is not actually leaked into another routing instance unless it is accepted by a leak-import policy of that other routing instance.

The **bgp-leak** command has an effect only when the policy is applied as a BGP import policy in the base router or a VPRN context.

**Default** **no default**

### community

**Syntax** **community** {{**add** [**remove**]} | {**remove** [**add**]} | {**replace**}}  
**no community**

**Context** config>router>policy-options>policy-statement>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command adds or removes a BGP community list to or from routes matching the route policy statement entry.

If no community list is specified, the community path attribute is not changed.

The community list changes the community path attribute according to the **add** and **remove** keywords.

The **no** form of the command disables the action to edit the community path attribute for the route policy entry.

**Default** **no community** — The community path attribute is not changed.

**Parameters** **add** — The specified community list is added to any existing list of communities.

**remove** — The specified community list is removed from the existing list of communities.

**replace** — The specified community list replaces any existing community attribute.

**name**—The community list name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

### damping

**Syntax** **damping** {*name* | **none**}  
**no damping**

**Context** config>router>policy-options>policy-statement >default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command configures a damping profile used for routes matching the route policy statement entry.

If no damping criteria is specified, the default damping profile is used.

The **no** form of the command removes the damping profile associated with the route policy entry.

**Default** **no damping** — Use the default damping profile.

**Parameters** *name* — The damping profile name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must be enclosed by at-signs (@) and may be midstring; for example, "@variable@", "start@variable@end", "@variable@end", or "start@variable@".

The *name* specified must already be defined.

**none** — Disables route damping for the route policy.

## fc

**Syntax** **fc** *fc* [**priority** {**low** | **high**}]  
**no fc**

**Context** config>router>policy-options>policy-statement>entry>action\$

**Description** This command associates a forwarding-class and optionally priority with the routes matched by a route policy entry. The command takes effect when the action of the route policy entry is **accept**, **next-entry** or **next-policy**. It has no effect except in route policies applied as VRF import policies, BGP import policies or RIP import policies.

The **no** form of the command removes the QoS association of the routes matched by the route policy entry.

**Default** **no fc**

**Parameters** *fc* — Specify the name of one of the predefined forwarding classes in the system.

**Values** be, l2, af, l1, h2, ef, h1, nc

**Default** none (no QoS information is associated with matched routes)

**priority** {**low** | **high**} — This parameter associates an enqueueing priority with routes matched by the policy entry. Specifying a priority is optional.

**Values** **high** — Setting the enqueueing parameter to **high** for a packet increases the likelihood of enqueueing the packet when the ingress queue is congested. Ingress enqueueing priority only affects ingress SAP queuing. Once the packet is placed in a buffer on the ingress queue, the significance of the enqueueing priority is lost.

**low** — Setting the enqueueing parameter to **low** for a packet decreases the likelihood of enqueueing the packet when the ingress queue is congested. Ingress enqueueing priority only affects ingress SAP queuing, once the packet is placed in a buffer on the ingress queue, the significance of the enqueueing priority is lost.

**Default** low

## default-action

**Syntax** **default-action** {**accept** | **next-entry** | **next-policy** | **reject**}  
**no default-action**

**Context** config>router>policy-options>policy-statement

**Description** This command enables the context to configure actions for routes that do not match any route policy statement entries when the **accept** parameter is specified.

## Route Policy Action Commands

The default action clause can be set to all available action states including: accept, reject, next-entry and next-policy. If the action states accept or reject then the policy evaluation terminates and the appropriate result is returned.

If a default action is defined and no match(es) occurred with the entries in the policy then the default action clause is used.

If a default action is defined and one or more matches occurred with the entries of the policy then the default action is not used.

The **no** form of the command deletes the **default-action** context for the policy statement.

**Default** **no default-action** — No default action is specified.

**Parameters**

- accept** — Specifies routes matching the entry match criteria will be accepted and propagated.
- next-entry** — Specifies that the actions specified would be made to the route attributes and then policy evaluation would continue with next policy entry (if any others are specified).
- next-policy** — Specifies that the actions specified would be made to the route attributes and then policy evaluation would continue with next route policy (if any others are specified).
- reject** — Specifies routes matching the entry match criteria would be rejected.

## install-backup-path

**Syntax** **install-backup-path**  
**no install-backup-path**

**Context** config>router>policy-options>policy-statement>entry>action  
config>router>policy-options>policy-statement>default-action

**Description** When the best BGP route for an IPv4 or IPv6 prefix is matched by a policy entry or policy default action with this command, BGP attempts to find and install a pre-programmed backup path for the prefix in order to provide BGP fast reroute protection.

The **install-backup-path** command overrides and has no dependency on commands such as the BGP instance **backup-path** command or the VPRN-level **enable-bgp-vpn-backup** command, which enable BGP fast reroute for an entire address family. The **install-backup-path** command provides more precise control over which IP prefixes are supported with pre-programmed backup paths.

If, within a VPRN, the best path for an IP prefix is provided by a VPRN BGP route, the backup path can be provided by another VPRN BGP route or an imported VPN-IP route. If, within a VPRN, the best path for an IP prefix is provided by an imported VPN-IP route, the backup path can be provided by another VPN-IP route.

The **install-backup-path** command is supported only in BGP import policies and VRF import policies and has no effect on policy types other than BGP import policies and VRF import policies. The **install-backup-path** command applies only to the following types of matched routes: IPv4, IPv6, label-IPv4, 6PE, VPN-IPv4, and VPN-IPv6.

**Default** **no default-action** — No default action is specified.



## local-preference

**Syntax** **local-preference** *preference*  
**no local-preference**

**Context** config>router>policy-options>policy-statement>default-action  
 config>router>policy-options>policy-statement>entry

**Description** This command assigns a BGP local preference to routes matching a route policy statement entry. If no local preference is specified, the BGP configured local preference is used. The **no** form of the command disables assigning a local preference in the route policy entry.

**Default** **No local-preference** — BGP default preference is assigned.

**Parameters** *preference* — The local preference expressed as a decimal integer.

**Values** 0 — 4294967295

*name*—The local preference parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

## metric

**Syntax** **metric** {**add**|**subtract**} *metric*  
**metric set** [**igp**|*metric-value*]  
**no metric**

**Context** config>router>policy-options>policy-statement>default-action  
 config>router>policy-options>policy-statement>entry>action

**Description** In a BGP import or export policy, this command assigns a MED value to routes matched by the policy statement entry. The MED value may be set to a fixed value (overriding the received value), set to the routing table cost of the route used to resolve the NEXT\_HOP of the BGP route (**igp** option), or modified by adding or subtracting a fixed value offset.

The **no** form of the command removes the MED attribute from the matched routes.

**Default** **no metric** — Uses the configured metric (if defined) or do not advertise a metric.

**Parameters** **add** — Specified *integer* is added to any existing metric. If the result of the addition results in a number greater than 4294967295, the value 4294967295 is used.

**subtract** — Specified *integer* is subtracted from any existing metric. If the result of the subtraction results in a number less than 0, the value of 0 is used.

**set** — Specified *integer* replaces any existing metric.

**igp** — Sets the MED value to the routing table cost of the route used to resolve the NEXT\_HOP of the BGP route.

## Route Policy Action Commands

*metric* — The metric modifier expressed as a decimal integer.

**Values** 0 — 4294967295

*name* — The metric parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”

### multicast-redirect

**Syntax** **multicast-redirect** [**fwd-service** *service-id*] *ip-int-name*  
**no multicast-redirect**

**Context** config>router>policy-options>policy-statement>default-action

**Description** This command enables a redirection under a filtering policy. The filtering policy in this case becomes a redirection policy and it is defined under the **router>policy-option** hierarchy.

Once the redirection policy is applied to the subscriber, all IGMP messages will be processed per subscriber host before they get redirected to the referenced interface (and possibly service). However, multicast traffic will not be replicated directly per subscriber host but instead it will be forwarded on the interface that is referenced in the redirection policy. The redirected interface must have IGMP enabled.

Currently all traffic is redirected and there is no ability to selectively redirect multicast traffic based on match conditions (multicast-groups, source IP address of IGMP messages, etc). Multicast redirection is supported between VPRN services and also between interfaces within the Global Routing Context. Multicast redirection is not supported between the VPRN services and the Global Routing Context. Multicast redirection is supported in the wholesale/retail VPRN context.

Note that when redirecting from a VPRN instance to the GRT is not supported. Redirecting from a VPRN to a different VPRN is supported and redirecting from an IES to another IES is also supported.

**Default** disabled

**Parameters** **fwd-service** *service-id* — Specifies the service to which traffic should be redirected. This option is applied only in the VPRN context. It is possible to redirect the multicast group into another service instance routing interface.

*ip-int-name* — specifies the alternate interface to which IGMP messages are redirected.

### next-hop

**Syntax** **next-hop** *ip-address*  
**no next-hop**

**Context** config>router>policy-options>policy-statement>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command assigns the specified next hop IP address to routes matching the policy statement entry.

If a next-hop IP address is not specified, the next-hop attribute is not changed.

The **no** form of the command disables assigning a next hop address in the route policy entry.

**Default** **no next-hop** — The next hop attribute is not changed.

**Parameters** *ip-address* — The next hop IP address in dotted decimal notation.

<b>Values</b>	ipv4-prefix:	a.b.c.d (host bits must be 0)
	ipv4-prefix-length:	0 — 32
	ipv6-prefix:	x:x:x:x:x:x:x (eight 16-bit pieces) x:x:x:x:x:d.d.d.d x: [0 — FFFF]H d: [0 — 255]D

*name*—The next-hop parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

## next-hop-self

**Syntax** [**no**] **next-hop-self**

**Context** config>router>policy-options>policy-statement *name*>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command advertises a next hop IP address belonging to this router even if a third-party next hop is available to routes matching the policy statement entry.

The **no** form of the command disables advertising the next-hop-self option for the route policy entry.

**Default** **no next-hop-self** — The next hop IP address is not changed.

## next-hop-self

**Syntax** [**no**] **next-hop-self** [**multihoming** *primary-anycast secondary-anycast*]

**Context** config>router>policy-option>policy-statement>entry>action

**Description** This command configures the group or neighbor to always set the NEXTHOP path attribute to its own physical interface when advertising to a peer. This is primarily used to avoid third-party route advertisements when connected to a multi-access network.

In addition, this command can be used to enable and configure the multi-homing reliency mechanism replacing the usual BGP nexthop with a configured anycast address.

The no form of the command returns the setting of the BGP next-hop attribute to the default value determined by the BGP protocol.

**Default** **no next-hop-self**

## Route Policy Action Commands

**Parameters** *primary-anycast* — Specifies the anycast address that the local node will use to replace the BGP nexthop address in route updates associated peers.  
*secondary-address* — Specifies the anycast address that the local node is to track.

### origin

**Syntax** **origin** {**igp** | **egp** | **incomplete** | *param-name*}  
**no origin**

**Context** config>router>policy-options>policy-statement *name*>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command sets the BGP origin assigned to routes exported into BGP.  
If the routes are exported into protocols other than BGP, this option is ignored.  
The **no** form of the command disables setting the BGP origin for the route policy entry.

**Default** **no origin**

**Parameters** **igp** — Sets the path information as originating within the local AS.

**egp** — Sets the path information as originating in another AS.

**incomplete** — Sets the path information as learned by some other means.

*param-name* — The origin parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, "@variable@".

### origin-validation-state

**Syntax** **origin-validation-state** *state*  
**no origin-validation-state**

**Context** config>router>policy-options>policy-statement *name*>default-action  
config>router>policy-options>policy-statement>entry>action

**Description** This command is used to mark BGP IPv4 and IPv6 routes matching the **default-action** or a specific entry of a route policy with one of the 3 following origin validation states:

- Valid (0)
- Not-Found (1)
- Invalid (2)

**Default** **no origin-validation-state**

**Parameters** **valid** — Marks the route as having an origin validation state of valid.

**notFound** — Marks the route as having an origin validation state of Not Found.

**invalid** — Marks the route as having an origin validation state of invalid.

## preference

**Syntax** **preference** *preference*  
**no preference**

**Context** config>router>policy-options>policy-statement *name*>default-action  
config>router>policy-options>policy-statement>entry>action>action

**Description** This command assigns a route preference to routes matching the route policy statement entry. If no preference is specified, the default Route Table Manager (RTM) preference for the protocol is used. The **no** form of the command disables setting an RTM preference in the route policy entry.

**Default** **no preference** — No route preference is assigned by the policy entry. The protocol default preference is used.

**Parameters** *preference* — The route preference expressed as a decimal integer.

**Values** 1 — 255 (0 represents unset - MIB only)

*name*—The preference parameter variable name . Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

## sticky-ecmp

**Syntax** **sticky-ecmp**  
**no sticky-ecmp**

**Context** config>router>policy-options>policy-statement>entry>action  
config>router>policy-options>policy-statement>default-action

**Description** This command specifies that BGP routes matching an entry or default-action of a route policy should be tagged internally as requiring sticky ECMP behavior. When a BGP route with multiple equal-cost BGP next-hops is programmed for sticky ECMP the failure of one or more of its BGP next-hops causes only the affected traffic flows to be re-distributed to the remaining next-hops; by default (without sticky-ECMP) all flows are potentially affected, even those using a next-hop that did not fail.

**Default** **no preference**

## Route Policy Action Commands

### tag

**Syntax** `tag tag`  
`no tag`

**Context** `config>router>policy-options>policy-statement>default-action`  
`config>router>policy-options>policy-statement>entry>action`

**Description** This command assigns a tag to routes matching the entry, which is then applied to IGP routes. A decimal or hexadecimal value of 4 octets can be entered.

For IS-IS and OSPF, all four octets can be used.

For RIP and RIPng, only the two most significant octets are used if more than two octets are configured.

The **no** form of the command removes the tag.

**Default** `no tag`

**Parameters** *tag* — Assigns an IS-IS, OSPF, RIP or RIPng tag to routes matching the entry.

**Values** Accepts decimal or hexadecimal formats:  
IS-IS and OSPF: 0x0–0xFFFFFFFF or 1–4294967295  
RIP and RIPng: 0x0–0xFFFF or 1–65535

*name* — The tag parameter variable name. Allowed values are any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within double quotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

### type

**Syntax** `type {type | param-name}`  
`no type`

**Context** `config>router>policy-options>policy-statement>default-action`  
`config>router>policy-options>policy-statement>entry>action`

**Description** This command sets the subtype for the Type 5 LSA (external LSA).

The **no** form of the command disables assigning a type in the route policy entry.

**Default** 2

**Parameters** *type* — Specifies the type metric.

**Values** Subtype 1— The external metric in the external LSA is comparable with the internal metric, and thus one can sum up all the metrics along the path (both internal and external) to get the total cost to the destination.

Subtype 2 — The metric in the external LSA is much more important than the internal

metric, so the internal metrics should only be considered when comparing two external routes that have the same external metric.

*name* — The type parameter variable name. Allowed values are any string upto 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, \$, spaces, etc.), the entire string must be enclosed within doublequotes. Policy parameters must start and end with at-signs (@); for example, “@variable@”.

## route-exists

**Syntax** **route-exists** *expression*  
**no route-exists**

**Context** config>router>policy-options>policy-statement>entry>cond-expr

**Description** This command is used to specify a route existence expression to control evaluation of the policy entry. If the route existence expression evaluates to ‘true’ the matching and action commands of the policy entry are applied as normal. If the route existence expression evaluates to ‘false’ the entire policy entry is skipped and processing continues with the next entry. Note however that conditional expressions are only parsed when the route policy is used as a BGP export policy or VRF export policy.

**Default** no route-exists

**Parameters** *expression* — “[“<prefix-list-name>”]” [all | none]

If neither the all nor the none keyword are used the match logic is ‘any’ – that is, the route expression evaluates as ‘true’ if any exact match entry in the referenced prefix-list has an active route in the route table associated with the policy.

**all** – the route expression evaluates as ‘true’ only if all the exact match entries in the referenced prefix-list have an active route in the route table associated with the policy.

**none** – the route expression evaluates as ‘true’ only if none of the exact match entries in the referenced prefix-list have an active route in the route table associated with the policy.

