Policy Configuration Guide

Version 21.6.1, 25 June 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

Table of Contents

1. Overview

Routing Policies are the rules that allow you to control and modify the default behaviour of the routing protocols such as BGP and IS-IS.

A routing policy consists of different "ordinals". These ordinals include "match" and "action" (with control) parts. The matched traffic with "match" field is acted according to the "action" field. For more information, see the Building Blocks of Policy Configuration section.

To use a routing policy, firstly you need to generate it. After this, you can enforce the policy to the routes.

1.1. Guidelines

- The names policies and policy lists can contain alphanumeric characters and underscore character. They must not include special characters like hyphen. For example, **BGP-EXPORT** is not supported, whereas **BGP_EXPORT** is supported. A valid name cannot start with a number but it can contain numbers and underscore (_) in the string. The length of the names should not exceed 64 characters.

1.2. Limitations

- Configuring the raw hardware package filters through a generic representation model is not supported
- The following features are unavailable in the current policy implementation:
  - Conditional policies, that is, filtering based on conditions (that is, if a route is present in table x, then permit or deny)
  - Filtering based on Layer 2 constructs like MAC and ARP
  - Access Control Lists on generic criteria
  - Integration of subscriber policy-based routing
  - Policy Main is not supported in this release

1.3. Policy Components

In RtBrick Full Stack, the policy is divided into 4 sub-components:

- Policy Repository
- Command Processing Module
- Policy Server, the policy generation and relationship management component
- Policy Client, the policy enforcement component
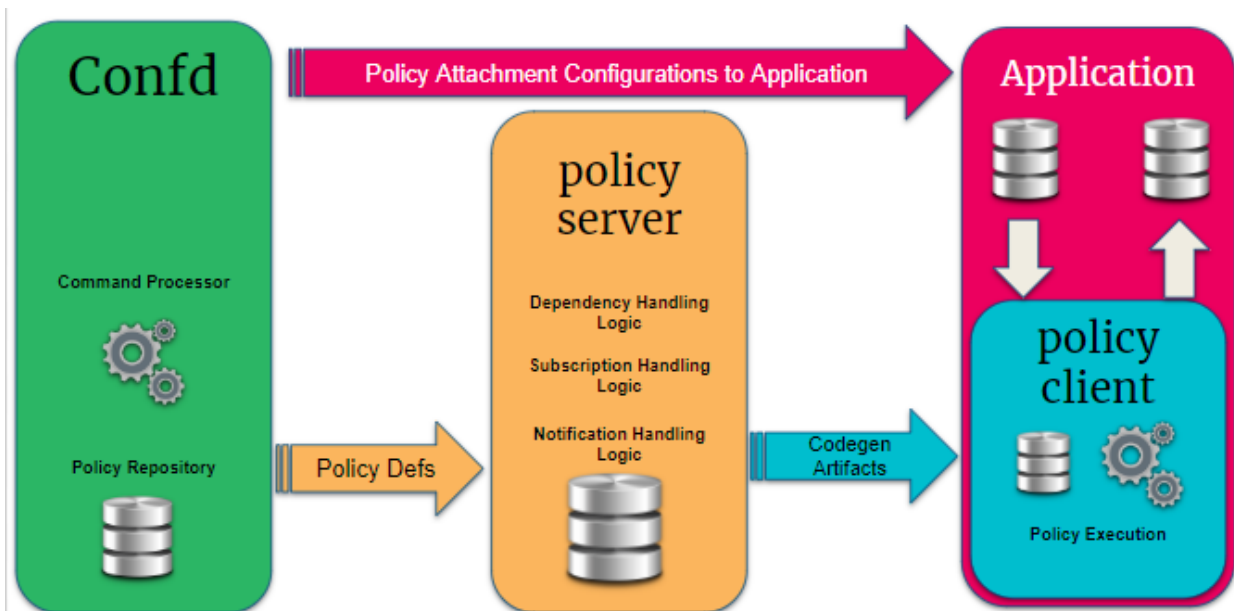
1.3.1. Policy Repository

The policy repository contains all the tables that are related to policy and associated list of compare criteria.

1.3.2. Command Processing Module

The command processing module is part of the Configuration daemon (confD), and it handles user interaction with the policy module. This is the back-end of the Command Line Interface (CLI) and JSON configuration that support the policy configurations.

This module maps the user-defined configuration into the back-end policy object, which is used by the execution engine (after verification), and it ensures that the policy can be correctly executed. In the future implementation of policy, this will be extended to include dynamic criteria for permit and deny actions that is based on routes present in a specific table. This module relays the user intent through Publish Subscribe bus (PubSub bus) to the policy server.

1.3.3. Policy Server

The Policy Server is a server component that manages all the policy rules in the various policy tables and also code generation of the policies.

The following are the functionalities of the policy server:

- Parses the objects in the policy tables, and it is an execution engine that generates the code to build the policy rules for evaluation, the relationship between various objects, and relays the intent to the evaluation engine.

- Maintains relationships between various policy constructs such as policy statements, rules, terms and lists.

- Tracks the attachment points so that when policies are modified, the appropriate clients are notified with the relevant new policies.

- Flattens the various relationships and generates a notification table that the clients subscribe to obtain notification based on specific interest groups.

- Uses the dependency table relationships to generate jobs to trigger code generation for various policy components.

- On code generation, the policy server updates a notification table that maintains the mapping between the policy server and the client interest groups. The notification table is a single point for the dissemination of information so that it can generate notifications for clients depending on their subscriptions for policy of interest.

- Policy server notification is generated for the policy clients. A notification is received from the notification table with metadata information that notifies the client if this is a new version of the policy or the original version of the policy. The client uses this information to enforce the policy evaluation and to decide on the version of the policy rule that should be used.

1.3.4. Policy Client

Policy client is a shared library component that a client daemon like BGP, IS-IS, OSPF etc. links to. This is the component that performs policy enforcement. It performs the following tasks:

- Links with client daemons like BGP, IS-IS, OSPF.

- Contains a listener that gets notifications on the availability of a new policy rule that is generated by the policy server.

- Evaluates the compiled rule and if there are any listeners/ interests, then notifies the components within the client daemon.

- Evaluates any policy configurations on the client daemon and invokes policy processing in response.

1.4. Support List Types

Following types of lists are supported:

- ipv4-prefix

- ipv4-address

- ipv6-prefix

- ipv6-address

- route-distinguisher

- community

- extended-community
- large-community
- as-path
- source
- sub-source
- cluster-list
- mpls-label
- mac-address

1.5. Building Blocks of Policy Configuration

The figure below shows the basic building blocks of the policy module. A policy is defined by a policy statement. A policy statement is a compound block of policy definition that consists of one or more policy terms.
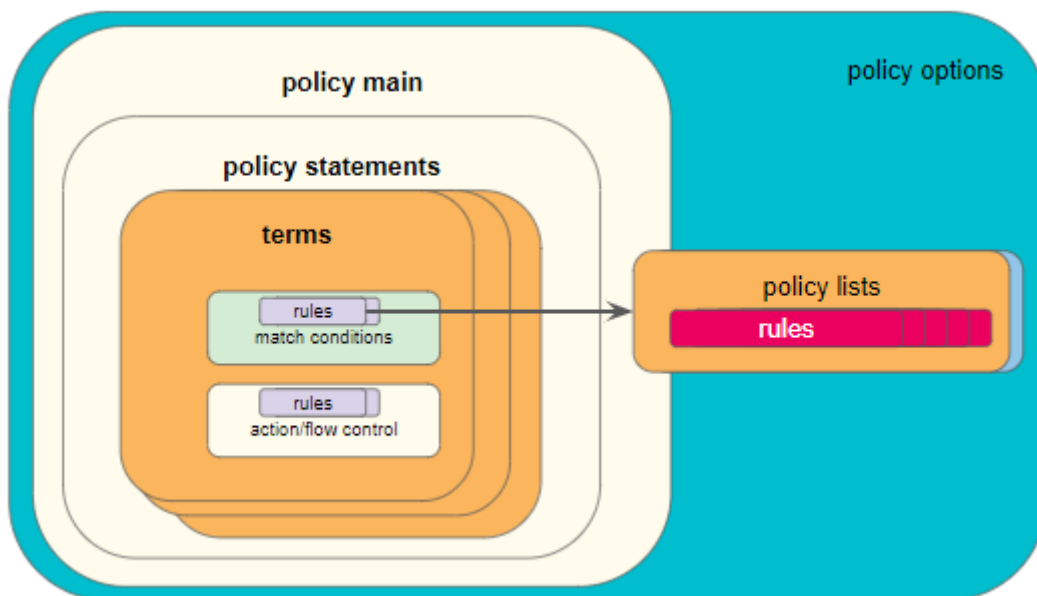
A policy term is the smallest block to relay user policy intent and consists of rules for match and action blocks. The match blocks can either define single independent elements like As-path, IP Prefix, IP addresses, Community, ext-community etc. or a list of the elements maintained in a different table.

Policy	Policy configuration mode
Policy Statement	Composed of one or more terms exercised in the order defined
Policy Ordinal	Ordinals are executed in ascending order.
Match	Match criteria to define either a single or list of independent elements. This is an optional block in a policy term

Action	Action criteria to either perform an action or define flow control, that is, go to next term, accept, deny etc. This is an optional block in policy term with a default action deny
Policy Lists	Aggregation or list of items of various types that are used in various policy comparison blocks
Rules	Index inside a policy term that defines the ordering of match and/or action criteria

1.5.1. Statement

A policy statement name is a globally unique string that is used to identify the policy and also used by the application for attachment points.

1.5.2. Ordinal

- An ordinal must be unique number within the scope of a statement which determines the order of the term execution within a policy statement.

- If no ordinal exist or configured, and if the policy is used, then all routes/BDS objects will be denied.

- Match must be associated to the match-condition, that is, and/or.

1.5.3. Match

- The outcome of a match block is deny by default.

- One or more matching rules make a match block; each matched routes/BDS objects are permitted by default.

- If a rule uses list match and if any one of the list entry matches to the attribute value, then the route is considered to be matched.

- If a list is defined and it is empty, then the route.bds object will be denied.

- Order of matching is based on the rule numbers.

- If match-type is or, then any one rule match will consider the route/BDS object as matched and permitted, otherwise it is denied.

- If match-type is and, then all rules match will consider the route/bds object as matched and permitted otherwise its denied.

- If match block results in a successful match, then corresponding action block is executed (resulting route/BDS object to be permitted).

- If match block results in a unsuccessful or there is no match, then corresponding action block is not executed instead next term is executed. If there are no more terms, then the policy execution will result in deny (resulting route/BDS object to be denied).

1.5.4. Actions in Policy

Action	Description
action goto-next-term	If next term exists, then next term is executed and the policy result is decided based on the result of the execution
action return-deny	Stops policy execution and returns result as deny (resulting route/BDS object to be denied)
action return-permit	Stops policy execution and return result as permit (resulting route/BDS object to be permitted)
operation delete-attribute	Deletes the attribute from the route/BDS object, that is, clearing all the info for that specific attribute in the object
operation <operation-type>	An operation is performed on the attribute in the route/BDS object based on the operation-type

1.6. Policy Match Options, Compare Types, and Operations

Application type	Operation Types Supported	Compare Types Supported
route-ipv4-prefix	overwrite	regex-match exact longer or-longer prefix-length-exact prefix-length-greater prefix-length-greater-or-exact
route-ipv6-prefix	overwrite	regex-match exact longer or-longer prefix-length-exact prefix-length-greater prefix-length-greater-or-exact
route-distinguisher	overwrite	regex-match exact
route-community	append prepend overwrite	regex-match exact exists
route-extended-community	append prepend overwrite	regex-match exact exists

Application type	Operation Types Supported	Compare Types Supported
route-large-community	append prepend overwrite	regex-match exact exists
route-as-path	append prepend overwrite	regex-match exact exists
route-cluster-list	append prepend overwrite	regex-match exact exists
route-source	overwrite	regex-match exact
route-sub-source	overwrite	regex-match exact
route-originator-identifier	overwrite	regex-match exact
route-peer-router-id	overwrite	regex-match exact
route-ipv4-nexthop	overwrite	regex-match exact
route-ipv6-nexthop	overwrite	regex-match exact
route-label	overwrite	regex-match exact
route-peer-ipv4	overwrite	regex-match exact
route-peer-ipv6	overwrite	regex-match exact
route-sid	overwrite	regex-match exact
route-sid-flag	overwrite	regex-match exact
route-external	overwrite	exact

1.6.1. Policy Compare Types

Policy Compare Types	Description
regex-match	An attribute can be matched using a standard Linux egrep regular expression. Example: "label": "label-op:push,label:206,bos:1" In this example, the label is a 64bit number, which has label value, bos, and operation encoding. A regex is used to match the string which is displayed in the table dump, that is, label-op:push,label:206,bos:1 not the 64bit value. The same is applicable to an array type attribute. A regex can be written to the string which is visible in the table dump output.
exact	Value configured in the command must be same as application attribute value
exists	This is applicable only for array type attribute; an exist match is the one where value configured in the command must exist in the application attribute value which is an array.
lesser	The application attribute value must be lesser than the value configured in the command
lesser-or-exact	The application attribute value must be lesser than or exact value configured in the command
greater	The application attribute value must be greater than the value configured in the command
greater-or-exact	The application attribute value must be greater than or exact value configured in the command
greater-longer	The route shares the same most-significant bits (described by prefix-length), and prefix-length is greater than the route's prefix length
greater-or-longer	The route shares the same most-significant bits (described by prefix-length), and prefix-length is equal to or greater than the route's prefix length.
longer	The route address shares the same most-significant bits as the match prefix (destination-prefix or source-prefix). The number of significant bits is described by the prefix-length component of the match prefix.

Policy Compare Types	Description
or-longer	The route address shares the same most-significant bits as the match prefix (destination-prefix orthe source-prefix). The number of significant bits is described by the prefix-length component of the match prefix.
prefix-length-exact	The application attribute value whose prefix length must be lesser than or exact value configured in the command
prefix-length-greater	The application attribute value whose prefix length must be greater than or exact value configured in the command

1.6.2. Policy Operation Types

Policy Operation Types	Description
Add	The application attribute value will be added with the value configured in the command
Subtract	The application attribute value will be subtracted with the value configured in the command. If the result of the subtraction results in a number less than 0, the value "0" is used.
Multiply	The application attribute value will be multiplied with the value configured in the command
Divide	The application attribute value will be divided with the value configured in the command
Overwrite	The application attribute value will be overwritten with the value configured in the command

1.7. Structure of Policy Statements

1.7.1. Syntax and Structure

The following example shows the syntax and structure of the policy statements.

```
supervisor@ixr_rtd: cfg> set policy
  <cr>
  list              Policy list configuration
  statement         Policy statement configuration
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 description "My
policy"
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match-condition or
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
```

```
   match-type            type                   value               value-type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
match-type
   exact                    Match type
   exists                   Match type
   greater                  Match type
   greater-or-exact         Match type
   less                     Match type
   less-or-exact            Match type
   longer                   Match type
   or-longer                Match type
   prefix-length-exact      Match type
   prefix-length-greater    Match type
   prefix-length-greater-or-exact  Match type
   regex-match              Match type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
match-type exact
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 type
   mcast-route-ipv4-group      Application type multicast route ipv4 group
   mcast-route-ipv4-source     Application type multicast route ipv4 source
   route-as-path               Application type route as path
   route-cluster-list          Application type route cluster list
   route-community             Application type route community
   route-distinguisher         Application type route distinguisher
   route-extended-community    Application type route extended community
   route-igp-metric            Application type route igp metric
   route-ipv4-nexthop          Application type route ipv4 nexthop
   route-ipv4-prefix           Application type route ipv4 prefix
   route-ipv6-nexthop          Application type route ipv6 nexthop
   route-ipv6-prefix           Application type route ipv6 prefix
   route-label                 Application type route label
   route-large-community       Application type route large community
   route-local-preference      Application type route local preference
   route-med                   Application type route med
   route-origin                Application type route origin
   route-originator-identifier Application type route originator identifier
   route-peer-ipv4             Application type route peer's ipv4 address
   route-peer-ipv6             Application type route peer's ipv6 address
   route-peer-router-id        Application type route peer's router
identifier
   route-preference            Application type route preference
   route-receive-path-identifier  Application type route receive path
identifier
   route-send-path-identifier  Application type route send path identifier
   route-source                Application type route source
   route-sub-source            Application type route sub source
   <type>                      Application type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 type
route-as-path
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
 match-type            type                   value               value-type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 value
   l1                   Application type value
   <value>             Application type value
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 value
10
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
 value-type
```

```
    complete            Value type
    list-name           Value type
supervisor@ixr_rtd: cfg> set policy
    <cr>
    list                Policy list configuration
    statement           Policy statement configuration
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1
 action              description         match               match-
condition
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 description "My
policy"
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match-condition or
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
 match-type           type                value               value-type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
match-type
    exact                       Match type
    exists                      Match type
    greater                     Match type
    greater-or-exact            Match type
    less                        Match type
    less-or-exact               Match type
    longer                      Match type
    or-longer                   Match type
    prefix-length-exact         Match type
    prefix-length-greater       Match type
    prefix-length-greater-or-exact  Match type
    regex-match                 Match type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
match-type exact
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 type
    mcast-route-ipv4-group      Application type multicast route ipv4 group
    mcast-route-ipv4-source     Application type multicast route ipv4 source
    route-as-path               Application type route as path
    route-cluster-list          Application type route cluster list
    route-community             Application type route community
    route-distinguisher         Application type route distinguisher
    route-extended-community    Application type route extended community
    route-igp-metric            Application type route igp metric
    route-ipv4-nexthop          Application type route ipv4 nexthop
    route-ipv4-prefix           Application type route ipv4 prefix
    route-ipv6-nexthop          Application type route ipv6 nexthop
    route-ipv6-prefix           Application type route ipv6 prefix
    route-label                 Application type route label
    route-large-community       Application type route large community
    route-local-preference      Application type route local preference
    route-med                   Application type route med
    route-origin                Application type route origin
    route-originator-identifier Application type route originator identifier
    route-peer-ipv4             Application type route peer's ipv4 address
    route-peer-ipv6             Application type route peer's ipv6 address
    route-peer-router-id        Application type route peer's router
identifier
    route-preference            Application type route preference
    route-receive-path-identifier  Application type route receive path
identifier
    route-send-path-identifier  Application type route send path identifier
    route-source                Application type route source
```

```
    route-sub-source                 Application type route sub source
    <type>                           Application type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 type
route-as-path
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
  match-type              type                    value                   value-type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 value
    <value>              Application type value
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1 value
10
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
value-type
    complete             Value type
    list-name            Value type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 match rule 1
value-type complete
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
  operation              type                    value
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
operation
    add                  Operation type
    append               Operation type
    delete-attribute     Operation type
    divide               Operation type
    goto-next-ordinal    Operation type
    multiply             Operation type
    overwrite            Operation type
    prepend              Operation type
    return-deny          Operation type
    return-permit        Operation type
    substract            Operation type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
operation prepend
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
    <cr>
    operation            Operation type
    type                 Application type
    value                Application type value
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1 type
  mcast-route-ipv4-group       Application type multicast route ipv4 group
  mcast-route-ipv4-source      Application type multicast route ipv4 source
  route-as-path                Application type route as path
  route-cluster-list           Application type route cluster list
  route-community              Application type route community
  route-distinguisher          Application type route distinguisher
  route-extended-community     Application type route extended community
  route-igp-metric             Application type route igp metric
  route-ipv4-nexthop           Application type route ipv4 nexthop
  route-ipv4-prefix            Application type route ipv4 prefix
  route-ipv6-nexthop           Application type route ipv6 nexthop
  route-ipv6-prefix            Application type route ipv6 prefix
  route-label                  Application type route label
  route-large-community        Application type route large community
  route-local-preference       Application type route local preference
  route-med                    Application type route med
  route-origin                 Application type route origin
  route-originator-identifier  Application type route originator identifier
  route-peer-ipv4              Application type route peer's ipv4 address
```

```
  route-peer-ipv6              Application type route peer's ipv6 address
  route-peer-router-id         Application type route peer's router
identifier
  route-preference             Application type route preference
  route-receive-path-identifier  Application type route receive path
identifier
  route-send-path-identifier   Application type route send path identifier
  route-source                 Application type route source
  route-sub-source             Application type route sub source
  <type>                       Application type
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1 type
route-community
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
value
 <value>
supervisor@ixr_rtd: cfg> set policy statement s1 ordinal 1 action rule 1
value 100:100
supervisor@ixr_rtd: cfg>
List-:
supervisor@ixr_rtd: cfg> set policy list
  l1                    List name
  <name>                List name
supervisor@ixr_rtd: cfg> set policy list l1
  as-path               Policy list type
  cluster-list          Policy list type
  community             Policy list type
  extended-community    Policy list type
  ipv4-address          Policy list type
  ipv4-prefix           Policy list type
  ipv6-address          Policy list type
  ipv6-prefix           Policy list type
  large-community       Policy list type
  mac-address           Policy list type
  mpls-label            Policy list type
  route-distinguisher   Policy list type
  source                Policy list type
  sub-source            Policy list type
  <type>                List type
supervisor@ixr_rtd: cfg> set policy list l1 as-path ordinal
  <ordinal>             Ordinal number
supervisor@ixr_rtd: cfg> set policy list l1 as-path ordinal 1
 <ordinal>
supervisor@ixr_rtd: cfg> set policy list l1 as-path ordinal 1 value
  <value>               List type value
supervisor@ixr_rtd: cfg> set policy list l1 as-path ordinal 1 value 10
```

1.7.2. Sample Configuration for Policy

```
set policy list l1 ipv4-prefix
set policy list l1 ipv4-prefix ordinal 1
set policy list l1 ipv4-prefix ordinal 1 value 200.0.0.1/32
set policy list l1 ipv4-prefix ordinal 2
set policy list l1 ipv4-prefix ordinal 2 value 200.0.0.3/32
set policy list l1 ipv4-prefix ordinal 3
set policy list l1 ipv4-prefix ordinal 3 value 200.0.0.5/32
set policy list l1 ipv4-prefix ordinal 4
set policy list l1 ipv4-prefix ordinal 4 value 200.0.0.7/32
set policy list l1 ipv4-prefix ordinal 5
set policy list l1 ipv4-prefix ordinal 5 value 200.0.0.9/32
set policy statement p1
set policy statement p1 ordinal 1
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 3
set policy statement p1 ordinal 1 match rule 3 type route-ipv4-prefix
set policy statement p1 ordinal 1 match rule 3 value-type complete
set policy statement p1 ordinal 1 match rule 3 match-type regex-match
set policy statement p1 ordinal 1 match rule 3 value "[0-9]{1,3}.[0-
9]{1,3}.[0-9]{1,3}.(15)/[0-9]{1,2}"
set policy statement p1 ordinal 1 action rule 1
set policy statement p1 ordinal 1 action rule 1 type route-local-preference
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 123
set policy statement p1 ordinal 1 action rule 3
set policy statement p1 ordinal 1 action rule 3 operation goto-next-ordinal
set policy statement p1 ordinal 2
set policy statement p1 ordinal 2 match-condition or
set policy statement p1 ordinal 2 match rule 1
set policy statement p1 ordinal 2 match rule 1 type route-ipv4-prefix
set policy statement p1 ordinal 2 match rule 1 value-type list-name
set policy statement p1 ordinal 2 match rule 1 match-type exact
set policy statement p1 ordinal 2 match rule 1 value l1
set policy statement p1 ordinal 2 action rule 1
set policy statement p1 ordinal 2 action rule 1 type route-med
set policy statement p1 ordinal 2 action rule 1 operation overwrite
set policy statement p1 ordinal 2 action rule 1 value 321
set policy statement p1 ordinal 3
set policy statement p1 ordinal 3 match-condition or
set policy statement p1 ordinal 3 match rule 3
set policy statement p1 ordinal 3 match rule 3 type route-ipv4-prefix
set policy statement p1 ordinal 3 match rule 3 value-type complete
set policy statement p1 ordinal 3 match rule 3 match-type regex-match
set policy statement p1 ordinal 3 match rule 3 value "[0-9]{1,3}.[0-
9]{1,3}.[0-9]{1,3}.(16)/[0-9]{1,2}"
set policy statement p1 ordinal 3 action rule 4
set policy statement p1 ordinal 3 action rule 4 type route-local-preference
set policy statement p1 ordinal 3 action rule 4 operation delete-attribute
set policy statement p1 ordinal 3 action rule 5
set policy statement p1 ordinal 3 action rule 5 operation return-deny
```

1.8. Tables and Subscriptions

The table below shows the various tables and their sharing across various policy

components.

Confd	global.policy.list.config global.policy.list.entry.config global.policy.match.rules.config global.policy.statement.config global.policy.ordinal.config global.policy.mapping.list global.policy.mapping.rules	Policy Statement is composed of one or more policy terms. Each term has a match action criteria. In the match and action criteria either a single element or a list of elements are compared and actions are taken. The actions include accept, deny, flow-control etc.
policy.server	global.policy.dependency global.<bds_name>.policy.subscription global.<bds_name>.policy.notification	Policy Server subscribes to all the tables from confd and creates tables that track policy-entry and dependency and notifies clients after code generation.
policy.client	global.<bds_name>.policy.shared.object.cache global.<bds_name>.policy.subscription global.<bds_name>.policy.context	Subscribes to code generation notifications, application context and maintains cache of subscribed .so

1.9. Using Policy with BGP

RtBrick supports attaching a BGP routing policy at two levels:

- Peer group address-family level

- Instance address-family level

In each case, you can apply the policy as an import or export policy and filter. As expected, import filters determine which routing updates are accepted and export filters determine which routes are advertised to other peers.

For more information, see the *RBFS BGP Configuration Guide*.

1.10. Using the Policy Test Feature

You can use the policy test feature to test a policy before attaching it to a BGP peer group or an instance.

Perform the following tasks:

1. Identify the table that you want to input to the policy.

2. Run the 'test policy' command on the table that you identified in step-1.

```
supervisor@leaf1: op> test policy run bgp.appd.1
    policy-name           Policy name
supervisor@leaf1: op> test policy run bgp.appd.1 policy-name s2 table
    bds.default.job.table
    bds_cmd_lib.default.job.table
    default.bgp.1.fib-local.ipv4.labeled-unicast
    default.bgp.1.fib-local.ipv4.multicast
     <....>
supervisor@leaf1: op> test policy run bgp.appd.1 policy-name s2 table
global.policy.bgp.appd.1.s
  global.policy.bgp.appd.1.shared.object.cache
global.policy.bgp.appd.1.statistics
global.policy.bgp.appd.1.subscription
supervisor@leaf1: op> test policy run bgp.appd.1 policy-name s2 table
global.policy.bgp.appd.1.shared.object.cache
supervisor@leaf1: op> test policy
    clear                 Clear a policy test result
    run                   Test a policy
supervisor@leaf1: op> test policy clear bgp.appd.1 policy-name
    <policy-name>
supervisor@leaf1: op> test policy clear bgp.appd.1 policy-name s2 table
global.policy.bgp.appd.1.shared.object.cache
```

1.10.1. Clearing the Result Tables

You can clear the result tables using the 'test policy clear' command:

```
supervisor@leaf1: op> test policy clear bgp.appd.1 policy-name s2 table
global.policy.bgp.appd.1.shared.object.cache
```

2. Policy Configuration Commands

This section presents a list of commands used for configuring policies. For information about the match options, compare operations and types, refer to Policy Match Options, Compare Types, and Operations.

2.1. Policy list rules configuration

2.1.1. ordinal value

To specify the order of rule execution, enter the following command:

Syntax

> **set policy list** <name> <type> **ordinal** <ordinal> **value** <value>

Command arguments

<name>	List name
<type>	Policy list type
<ordinal>	it gives order of execution of rules
<value>	Mention proper value

Example

```
set policy list l1 ipv4-prefix ordinal 1 value 200.0.0.1/32
set policy list l1 ipv4-prefix ordinal 2 value 200.0.0.3/32
set policy list l1 ipv4-prefix ordinal 3 value 200.0.0.5/32
set policy list l1 ipv4-prefix ordinal 4 value 200.0.0.7/32
set policy list l1 ipv4-prefix ordinal 5 value 200.0.0.9/32
```

To delete the operation that you performed, enter the following command:

delete ordinal <ordinal>

Command arguments

<ordinal>	it gives order of deleting the executed rules

Example

```
delete policy list l1 ipv4-prefix ordinal 1
delete policy list l1 ipv4-prefix ordinal 2
delete policy list l1 ipv4-prefix ordinal 3
delete policy list l1 ipv4-prefix ordinal 4
delete policy list l1 ipv4-prefix ordinal 5
```

2.2. Policy rules match list options

2.2.1. ipv4-prefix match-list

Use this command to set the rule to match for ipv4 prefix list

Syntax

set rule <rule> **route ipv4-prefix match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv4-prefix
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.2. ipv6-prefix match-list

Use this command to set the rule to match for ipv6 prefix list

Syntax

set rule <rule> **route ipv6-prefix match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy

<list-name>	name of the list of prefix which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv6-prefix
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.3. distinguisher match-list

Use this command to set the rule to match for distinguisher list

Syntax

set rule <rule> **route distinguisher match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-distinguisher
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.4. community match-list

Use this command to set the rule to match for community list

Syntax

set rule <rule> **route community match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-community
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.5. extended-community match-list

Use this command to set the rule to match for extended-community list

Syntax

set rule <rule> **route extended-community match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-extended-community
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.6. large-community match-list

Use this command to set the rule to match for large-community list

Syntax

set rule <rule> **route large-community match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-large-community
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.7. as-path match-list

Use this command to set the rule to match for as-path list

Syntax

set rule <rule> **route as-path match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-as-path
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
exit
```

2.2.8. cluster-list match-list

Use this command to set the rule to match for cluster-list

Syntax

set rule <rule> **route cluster-list match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-cluster-list
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.9. source match-list

Use this command to set the rule to match for source list

Syntax

set rule <rule> **route source match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-source
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.10. sub-source match-list

Use this command to set the rule to match for sub-source list

Syntax

set rule <rule> **route sub-source match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-sub-source
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.11. originator-identifier match-list

Use this command to set the rule to match for originator-identifier list

Syntax

set rule <rule> **route originator-identifier match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-riginator-
identifier
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.12. peer-router-id match-list

Use this command to set the rule to match for peer_router-id list

Syntax

set rule <rule> **route peer-router-id match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-router-id
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.13. ipv4-nexthop match-list

Use this command to set the rule to match for ipv4-nexthop list

Syntax

set rule <rule> **route ipv4-nexthop match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv4-nexthop
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.14. ipv6-nexthop match-list

Use this command to set the rule to match for ipv6-nexthop list

Syntax

set rule <rule> **route ipv6-nexthop match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv6-nexthop
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.15. label match-list

Use this command to set the rule to match for label list

Syntax

set rule <rule> **route label match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-label
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.16. peer-ipv4 match-list

Use this command to set the rule to match for peer-ipv4 list

Syntax

set rule <rule> **route peer-ipv4 match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-ipv4
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.2.17. peer-ipv6 match-list

Use this command to set the rule to match for peer-ipv6 list

Syntax

set rule <rule> **route peer-ipv6 match-list-name** <list-name>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<list-name>	name of the list of prefix or attributes which is going to be used for setting up the policy

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-ipv6
set policy statement p1 ordinal 1 match rule 1 value-type list-name
set policy statement p1 ordinal 1 match rule 1 value l1
```

2.3. Policy rules delete for match

2.3.1. delete rule

Use this command to delete the rule set for policy statement

Syntax

delete rule <rule>

Command arguments

<rule>	Specifies the rule that you want to delete

Example

```
delete policy statement p1 ordinal 1 match rule 1
```

2.4. Policy rules match rules options

2.4.1. ipv4-prefix match-type

Use this command to setup a rule to match-value type for route ipv4-prefix

Syntax

set rule <rule> **route ipv4-prefix match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv4 address, for example 2.2.2.4/24

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type prefix-length-exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv4-prefix
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 2.2.2.4/24
```

2.4.2. ipv6-prefix match-type

Use this command to setup a rule to match-value type for route ipv6-prefix

Syntax

set rule <rule> **route ipv6-prefix match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv6 address, for example 2001:db8:3c4d:15::/64

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv6-prefix match
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 2001:db8:3c4d:15::/64
```

2.4.3. distinguisher match-type

Use this command to setup a rule to match-value type for route distinguisher

Syntax

set rule <rule> **route distinguisher match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route distinguisher value, for example 192.168.1.1:65002

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-distinguisher
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 192.168.1.1:65002
```

2.4.4. community match-type

Use this command to setup a rule to match-value type for route community

Syntax

set rule <rule> **route community match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route community value, for example 7018:5000

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-community
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 7018:5000
```

2.4.5. extended-community match-type

Use this command to setup a rule to match-value type for route extended-community

Syntax

set rule <rule> **route extended-community match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route extended-community value, for example 192.168.0.0:5000

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-extended-community
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 192.168.0.0:5000
```

2.4.6. large-community match-type

Use this command to setup a rule to match-value type for route larger-community

Syntax

set rule <rule> **route large-community match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route large-community value, for example 2914:65400:5000

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-large-community
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 2914:65400:5000
```

2.4.7. as-path match-type

Use this command to setup a rule to match-value type for route as-path

Syntax

set rule <rule> **route as-path match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route as-path value, for example 65001

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-as-path
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 65001
```

2.4.8. cluster-list match-type

Use this command to setup a rule to match-value type for route cluster-list

Syntax

set rule <rule> **route cluster-list match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv4 address, for example 10.10.10.2

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-cluster-list
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 192.168.1.5
```

2.4.9. origin match-type

Use this command to setup a rule to match-value type for route origin

Syntax

set rule <rule> **route origin match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route origin value, for example IGP, EGP etc

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-origin
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value IGP
```

2.4.10. med match-type

Use this command to setup a rule to match-value type for route MED

Syntax

set rule <rule> **route med match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy

<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route med value, for example 100,200 etc

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-med
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 100
```

2.4.11. local-preference match-type

Use this command to setup a rule to match-value type for local-preference route

Syntax

set rule <rule> **route local-preference match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route local-preference value, for example 100, 200 etc

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-local-preference
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 100
```

2.4.12. preference match-type

Use this command to setup a rule to match-value type for preference route

Syntax

set rule <rule> **route preference match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route preference value, for example 2, 100, 200 etc

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-preference
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 100
```

2.4.13. source match-type

Use this command to setup a rule to match-value type for source route

Syntax

set rule <rule> **route source match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route source value, for example BGP

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-source
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value bgp
```

2.4.14. sub-source match-type

Use this command to setup a rule to match-value type for sub-source route

Syntax

set rule <rule> **route sub-source match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route sub-source value, for example 100

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-sub-source
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 200
```

2.4.15. originator-identifier match-type

Use this command to setup a rule to match-value type for originator-identifier route

Syntax

set rule <rule> **route originator-identifier match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy

<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv4 address, for example 20.20.20.4

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-originator-id
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 192.168.1.7
```

2.4.16. peer-router-id match-type

Use this command to setup a rule to match-value type for peer-router-id

Syntax

set rule <rule> **route peer-router-id match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv4 address, for example 20.20.20.4

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-router-id
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 5.1.168.192
```

2.4.17. ipv4-nexthop match-type

Use this command to setup a rule to match-value type for ipv4-nexthop route

Syntax

set rule \<rule> **route ipv4-nexthop match** \<match-type> **value** \<attribute-value>

Command arguments

\<rule>	rule which are going to set either for match or action for policy
\<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
\<attribute-value>	Specify a valid IPv4 address, for example 20.20.20.4

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv4-nexthop
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 129.121.76.192
```

2.4.18. ipv6-nexthop match-type

Use this command to setup a rule to match-value type for ipv6-nexthop route

Syntax

set rule \<rule> **route ipv6-nexthop match** \<match-type> **value** \<attribute-value>

Command arguments

\<rule>	rule which are going to set either for match or action for policy
\<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
\<attribute-value>	Specify a valid IPv6 address, for example 17f0:949f:6a53:898f:8369:beb9:cd89:5ced

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-ipv6-nexthop
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value
17f0:949f:6a53:898f:8369:beb9:cd89:5ced
```

2.4.19. receive-path-identifier match-type

Use this command to setup a rule to match-value type for receive-path-identifier route

Syntax

set rule <rule> **route receive-path-identifier match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route receive-path-identifier value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-receive-path-id
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 1885178186
```

2.4.20. send-path-identifier match-type

Use this command to setup a rule to match-value type for send-path-identifier route

Syntax

set rule <rule> **route send-path-identifier match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route send-path-identifier value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-send-path-id
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 1885178186
```

2.4.21. label match-type

Use this command to setup a rule to match-value type for label route

Syntax

set rule <rule> **route label match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid mpls-lable, for example label-op:push,label:20001,bos-op:compare,bos:1

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-label
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value label-
op:push,label:20001,bos-op:compare,bos:1
```

2.4.22. igp-metric match-type

Use this command to setup a rule to match-value type for igp-metric route

Syntax

set rule <rule> **route igp-metric match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid route igp-metric value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-igp-metric
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 1885178186
```

2.4.23. peer-ipv4 match-type

Use this command to setup a rule to match-value type for peer-ipv4 route

Syntax

set rule <rule> **route peer-ipv4 match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv4 address, for example 129.121.76.192

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-ipv4
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value 129.121.76.192
```

2.4.24. peer-ipv6 match-type

Use this command to setup a rule to match-value type for peer ipv6-route

Syntax

set rule <rule> **route peer-ipv6 match** <match-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<match-type>	is condition of "and/or" ('and' means set of rules to be satisfied & 'or' means any one rule is satisfied policy should get applied)
<attribute-value>	Specify a valid IPv6 address, for example 17f0:949f:6a53:898f:8369:beb9:cd89:5ced

Example

```
set policy statement p1 ordinal 1 match-condition or
set policy statement p1 ordinal 1 match rule 1 match-type exact
set policy statement p1 ordinal 1 match rule 1 type route-peer-ipv6
set policy statement p1 ordinal 1 match rule 1 value-type complete
set policy statement p1 ordinal 1 match rule 1 value
17f0:949f:6a53:898f:8369:beb9:cd89:5ced
```

2.5. Policy rules action-rules options

2.5.1. ipv4-prefix operation

Use this command to perform the operation on configured ipv4-prefix rule

Syntax

set rule <rule> **route ipv4-prefix operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 20.20.20.3/24

Example

```
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 type route-ipv4-prefix
set policy statement p1 ordinal 1 action rule 1 value 1.1.1.1/24
```

2.5.2. ipv6-prefix operation

Use this command to perform the operation on configured ipv6-prefix rule

Syntax

set rule <rule> **route ipv6-prefix operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv6 address, for example 2001:db8:3c4d:15::/64

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-ipv6-prefix
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 2001:db8:3c4d:15::/64
```

2.5.3. distinguisher operation

Use this command to perform the operation on configured distinguisher rule

Syntax

set rule <rule> **route distinguisher operation** <operation-type> **value** <attribute-

value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-distinguisher
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 192.168.1.4:65002
```

2.5.4. community operation

Use this command to perform the operation on configured community rule

Syntax

set rule <rule> **route community operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route community value, for example 7018:5000

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-community
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 7018:5000
```

2.5.5. extended-community operation

Use this command to perform the operation on configured extended-community

rule

Syntax

set rule <rule> **route extended-community operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route extended-community value, for example 192.168.0.0:5000

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-extended-community
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 192.168.0.0:5000
```

2.5.6. large-community operation

Use this command to perform the operation on configured large-community rule

Syntax

set rule <rule> **route large-community operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route large-community value, for example 2914:65400:5000

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-large-community
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 2914:65400:5000
```

2.5.7. as-path operation

Use this command to perform the operation on configured as-path rule

Syntax

set rule <rule> **route as-path operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route as-path value, for example 65001

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-as-path
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 65002
```

2.5.8. cluster-list operation

Use this command to perform the operation on configured cluster-list rule

Syntax

set rule <rule> **route cluster-list operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 52.10.100.250

Example

```
edit policy-options
edit policy-statement p1
edit term t1 ordinal 1
edit match-conditions match-type or
edit rules
set rule 1 route cluster-list operation overwrite value 52.10.100.250
exit
exit
exit
exit
exit
```

2.5.9. origin operation

Use this command to perform the operation on configured origin rule

Syntax

set rule <rule> **route origin operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route origin value, for example IGP, EGP etc

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-origin
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 123
```

2.5.10. med operation

Use this command to perform the operation on configured MED rule

Syntax

set rule <rule> **route med operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route med value, for example 100,200 etc

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-med-operation
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 123
```

2.5.11. local-preference operation

Use this command to perform the operation on configured local-preference rule

Syntax

set rule <rule> **route local-preference operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route local-preference value, for example 100, 200 etc

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-local-preference
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 123
```

2.5.12. preference operation

Use this command to perform the operation on configured preference rule

Syntax

set rule <rule> **route preference operation** <operation-type> **value** <attribute-

value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route preference value, for example 2, 100, 200 etc

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-preference
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 123
```

2.5.13. source operation

Use this command to perform the operation on configured source rule

Syntax

set rule <rule> **route source operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route source value, for example BGP

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-source
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value bgp
```

2.5.14. sub-source operation

Use this command to perform the operation on configured sub-source rule

Syntax

set rule <rule> **route sub-source operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route sub-source value, for example 100

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-sub-source
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 303243
```

2.5.15. originator-identifier operation

Use this command to perform the operation on configured originator-identifier rule

Syntax

set rule <rule> **route originator-identifier operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 129.121.76.192

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-originator-
identifier
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 192.168.2.8
```

2.5.16. peer-router-id operation

Use this command to perform the operation on configured peer-router-id rule

Syntax

set rule <rule> **route peer-router-id operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 129.121.76.192

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-peer-router-id
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 5.1.168.192
```

2.5.17. ipv4-nexthop operation

Use this command to perform the operation on configured ipv4-nexthop rule

Syntax

set rule <rule> **route ipv4-nexthop operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 129.121.76.192

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-ipv4-nexthop
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 10.10.10.2
```

2.5.18. ipv6-nexthop operation

Use this command to perform the operation on configured ipv6-nexthop rule

Syntax

set rule <rule> **route ipv6-nexthop operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv6 address, for example 17f0:949f:6a53:898f:8369:beb9:cd89:5ced

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-ipv4-nexthop
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value
17f0:949f:6a53:898f:8369:beb9:cd89:5ced
```

2.5.19. receive-path-identifier operation

Use this command to perform the operation on configured receive-path-id rule

Syntax

set rule <rule> **route receive-path-identifier operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type

<attribute-value>	Specify a valid route receive-path-identifier value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-receive-path-
identifier
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 1885178186
```

2.5.20. send-path-identifier operation

Use this command to perform the operation on configured send-path-id rule

Syntax

set rule <rule> **route send-path-identifier operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route send-path-identifier value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-send-path-
identifier
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 1885178186
```

2.5.21. label operation

Use this command to perform the operation on configured label rule

Syntax

set rule <rule> **route label operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid mpls-lable, for example label-op:push,label:20001,bosop:compare,bos:1

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-label operation
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value label-op
```

2.5.22. igp-metric operation

Use this command to perform the operation on configured igp-metric rule

Syntax

set rule <rule> **route igp-metric operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid route igp-metric value, for example 1885178186

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-igp-metric
operation
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 12323
```

2.5.23. peer-ipv4 operation

Use this command to perform the operation on configured peer-ipv4 rule

Syntax

set rule <rule> **route peer-ipv4 operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv4 address, for example 129.121.76.192

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-peer-ipv4
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value 26.0.1.1
```

2.5.24. peer-ipv6 operation

Use this command to perform the operation on configured peer-ipv6 rule

Syntax

set rule <rule> **route peer-ipv6 operation** <operation-type> **value** <attribute-value>

Command arguments

<rule>	rule which are going to set either for match or action for policy
<operation-type>	An operation is performed on that attribute in the route/BDS object based on the operation-type
<attribute-value>	Specify a valid IPv6 address, for example 17f0:949f:6a53:898f:8369:beb9:cd89:5ced

Example

```
set policy statement p1 ordinal 1 action rule 1 type route-peer-ipv6
set policy statement p1 ordinal 1 action rule 1 operation overwrite
set policy statement p1 ordinal 1 action rule 1 value
17f0:949f:6a53:898f:8369:beb9:cd89:5ced
```