

Chapter 10

Configure Routing Tables and Routes

This chapter describes the following tasks for creating and configuring routing tables:

- Create Routing Tables on page 91
- Configure Static Routes on page 93
- Configure Aggregate Routes on page 101
- Configure Generated Routes on page 108
- Configure Martian Addresses on page 114

Create Routing Tables

The JUNOS software can maintain one or more routing tables, thus allowing the software to store route information learned from different protocols separately. For example, it is common for the routing software to maintain unicast routes and multicast routes in different routing tables. You also might have policy considerations that would lead you to create separate routing tables to manage the propagation of routing information.

Creating routing tables is optional. If you do not create any, the JUNOS software uses its default routing tables, which are `inet.0` for unicast routes, `inet.1` for the multicast forwarding cache, and `inet.3` for Multiprotocol Label Switching (MPLS). If Multiprotocol Border Gateway Protocol (MBGP) is enabled, `inet.2` is used for subaddress family indicator (SAFI) 2 routes.

If you want to add static, aggregate, generated, or martian routes only to the default unicast routing table (`inet.0`), you do not have to create any routing tables because, by default, these routes are added to `inet.0`. You can add these routes just by including the `static`, `aggregate`, `generate`, and `martians` statements at the `[edit routing-options]` hierarchy level.

To explicitly create a routing table, include the `rib` statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    static {
      defaults {
        static-options;
      }
      route destination-prefix {
        next-hop;
        static-options;
      }
    }
  }
  aggregate {
    defaults {
      aggregate-option;
    }
    route destination-prefix {
      policy policy-name;
      aggregate-options;
    }
  }
  generate {
    defaults {
      generate-options;
    }
    route destination-prefix {
      policy policy-name;
      generate-options;
    }
  }
  martians {
    destination-prefix match-type <allow>;
  }
}
}
```

The routing table name, *routing-table*, consists of the protocol family, optionally followed by a period and a number. The protocol family can be `inet` for the IP family or `iso` for the ISO protocol family. The number represents the instance of the routing table. The first instance is 0.

Example: Create a Routing Table

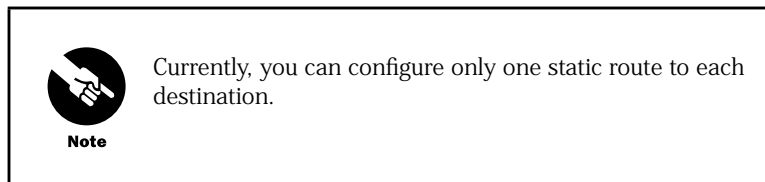
Create the routing table `inet.4` and add a static route to it:

```
[edit]
routing-options {
  rib inet.4 {
    static {
      route 140.122.0.0/16 next-hop 192.168.0.10;
    }
  }
}
```

Configure Static Routes

The router uses dynamic routes to learn how to reach network destinations. Dynamic routes are determined from the information exchanged by the routing protocols and, as the name implies, the routes might change as network conditions change and these changes are discovered by the routing protocols. You can configure static (nonchanging) routes to some network destinations. The router uses static routes when it does not have a route to a destination that has a better (lower) preference value, when it cannot determine the route to a destination, or when it is forwarding unroutable packets.

A static route is installed in the routing table only when the route is active; that is, the list of next-hop routers configured for that route contains at least one next hop on an operational interface.



You can add the same routes to more than one routing table.

To configure static routes in the default routing table (inet.0), include the static statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  static {
    defaults {
      static-options;
    }
    route destination-prefix {
      next-hop;
      static-options;
    }
  }
}
```

To configure static routes in one of the other routing tables, or to explicitly configure static routes in the default route table (inet.0), include the static statement at the [edit routing-options rib *routing-table*] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    static {
      defaults {
        static-options;
      }
      route destination-prefix {
        next-hop;
        static-options;
      }
    }
  }
}
```

The static statement consists of two parts:

defaults—Specify global static route options. These are treated as global defaults and apply to all the static routes you configure in the static statement. This part of the static statement is optional.

route—Configure individual static routes. In this part of the static statement, you optionally can configure static route options. These options apply to the individual destination only and override any options you configured in the defaults part of the static statement.

The following sections explain how to configure static routes:

Specify the Destination of the Static Route on page 94

Specify the Next Hop of the Static Route on page 94

Specify Static Route Options on page 95

Configure a Default Route on page 99

Propagate Static Routes into Routing Protocols on page 99

Examples: Configure Static Routes on page 100

Specify the Destination of the Static Route

When you configure an individual static route in the route part of the static statement, specify the destination of the route (in route *destination-prefix*) in one of the following ways:

network/masklen, where *network* is the network portion of the IP address and *masklen* is the destination prefix length.

default if this is the default route to the destination. This is equivalent to specifying an IP address of 0.0.0.0/0.

Specify the Next Hop of the Static Route

When you configure an individual static route in the route part of the static statement, specify how to reach the destination (in *next-hop*) in one of the following ways:

next-hop address—IP address of the next hop to the destination, specified as:

IP address of the next hop

Interface name (for point-to-point interfaces only)

address/interface-name to specify an IP address on an operational interface

reject—Do not forward packets addressed to this destination. Instead, drop the packets, send ICMP unreachable messages to the packets' originators, and install a reject route for this destination into the routing table.

discard—Do not forward packets addressed to this destination. Instead, drop the packets, do not send ICMP unreachable messages to the packets' originators, and install a reject route for this destination into the routing table.

receive—Cause packets to the destination to be received by the local router.

Specify Static Route Options

In the defaults and route parts of the static statement, you can specify *static-options*, which define additional information about static routes that is included with the route when it is installed in the routing table. All static options are optional. Static options that you specify in the defaults part of the static statement are treated as global defaults and apply to all the static routes you configure in the static statement. Static options that you specify in the route part of the static statement override any global static options and apply to that destination only.

To configure static route options, include one or more of them in the defaults or route part of the static statement. Each of these options is explained in the sections that follow.

```
[edit]
routing-options {
  static {
    (defaults | route) {
      (active | passive);
      as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
        <aggregator as-number in-address>;
      community ([community-ids] | no-advertise | no-export |
        no-export-subconfed | none);
      (install | no-install);
      metric metric <type type>;
      (preference | preference2 | color | color2) preference;
      (readvertise | no-readvertise);
      (no-retain | retain);
      (tag | tag2) string;
    }
  }
}
```

The following sections explain how to specify static route options:

Specify the Route Metric on page 96

Specify the Route Preference on page 96

Specify Community Information on page 96

Specify the AS Path on page 97

Specify the OSPF Tag on page 97

Specify Whether the Route Is Installed in the Forwarding Table on page 97

Specify Whether the Route Is Permanently Installed in the Forwarding Table on page 98

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table on page 98

Specify When the Route Can Be Readvertised on page 99

Specify the Route Metric

To associate a metric value with a route, include the metric statement at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
metric metric <type type>;
```

In the type option, you can specify the type of route. For OSPF, when routes are exported to OSPF, type 1 routes are advertised in type 1 externals and routes of any other type are advertised in type 2 externals.

Specify the Route Preference

By default, static routes have a preference value of 5. To modify the default preference value, specify a primary preference value (preference). You also can specify a secondary preference value (preference2); and colors, which are even finer-grained preference values (color and color2). To do this, include one or more of the following statements at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
(preference | preference2 | color | color2) preference;
```

The preference value can be a number in the range 1 through 255, with a lower number indicating a more preferred route. For more information about preference values, see “Route Preferences” on page 5.

Specify Community Information

By default, no BGP community information is associated with static routes. To associate community information with the routes, include the community option at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
community ([ community-ids ] | no-advertise | no-export | no-export-subconfed | none);
```

community-ids is one or more community identifiers in the format *as-number:community-value*. *as-number* is the AS number and can be a value in the range 1 through 65534. *community-value* is the community identifier and can be a number in the range 0 through 65535.

You also can specify *community-ids* as one of the following well-known community names, which are defined in RFC 1997:

no-export—Routes containing this community name are not advertised outside a BGP confederation boundary.

no-advertise—Routes containing this community name are not advertised to other BGP peers.

no-export-subconfed—Routes containing this community name are not advertised to external BGP peers, including peers in other members’ ASs inside a BGP confederation.

Specify the option **none** to explicitly exclude BGP community information with a static route. Include this option when configuring an individual route in the route portion of the static statement to override a community option specified in the defaults portion of the statement.

Specify the AS Path

By default, no AS path information is associated with static routes. To associate AS path information with the routes, include the `as-path` option at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
  <aggregator as-number in-address>;
```

`as-path` is the AS path to include with the route. It can include a combination of individual AS path numbers and AS sets. Enclose sets in brackets ([]). The first AS number in the path represents the AS immediately adjacent to the local AS. Each subsequent number represents an AS that is progressively farther from the local AS, heading toward the origin of the path.

You also can specify the AS path using the BGP origin attribute, which indicates the origin of the AS path information:

`igp`—Path information originated within the local AS.

`egp`—Path information originated in another AS.

`incomplete`—Path information learned by some other means.

To attach the BGP `ATOMIC_AGGREGATE` path attribute to the static route, specify the `atomic-aggregate` option. This path attribute indicates that the local system selected a less specific route rather than a more specific route.

To attach the BGP `AGGREGATOR` path attribute to the static route, specify the `aggregator` option. When using this option, you must specify the last AS number that formed the static route (encoded as two octets), followed by the IP address of the BGP system that formed the static route.

Specify the OSPF Tag

By default, no OSPF tag strings are associated with static routes. You can specify up to two OSPF tag strings by including the `tag` and `tag2` options at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
(tag | tag2) string;
```

Specify Whether the Route Is Installed in the Forwarding Table

By default, the JUNOS software installs all active static routes into the forwarding table. To configure the software not to install active static routes into the forwarding table, include the `no-install` option when configuring the route:

```
[edit routing-options static (defaults | route)]
no-install;
```

Even if you configure a route so it is not installed in the forwarding table, the route is still eligible to be exported from the routing table to other protocols. To explicitly install the routes into the forwarding table, include the `install` option when configuring routes. Include this option when configuring an individual route in the route portion of the static statement to override a `no-install` option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
install;
```

Specify Whether the Route Is Permanently Installed in the Forwarding Table

By default, statically configured routes are deleted from the forwarding table when the routing protocol process shuts down normally. To have a static route remain in the forwarding table, include the `retain` option when configuring the route. Doing this greatly reduces the time required to restart a system that has a large number of routes in its routing table.

```
[edit routing-options static (defaults | route)]
retain;
```

To explicitly specify that routes are deleted from the forwarding table, include the `no-retain` option when configuring routes. Include this option when configuring an individual route in the route portion of the static statement to override a `retain` option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
no-retain;
```

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table

By default, static routes are removed from the routing and forwarding tables when they become inactive. To have a static route remain continually installed in the routing and forwarding tables, include the `passive` option when configuring the route:

```
[edit routing-options static (defaults | route)]
passive;
```

Routes that have been configured to remain continually installed in the routing and forwarding tables are marked with `reject next hops` when they are inactive.

To explicitly remove static routes when they become inactive, include the `active` option when configuring routes. Include this option when configuring an individual route in the route portion of the static statement to override a `retain` option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
active;
```

Specify When the Route Can Be Readvertised

By default, static routes are eligible to be readvertised (that is, exported) by dynamic routing protocols. To mark a static route as being ineligible for readvertisement, include the `no-readvertise` option when configuring the route:

```
[edit routing-options static (defaults | route)]
no-readvertise;
```

To explicitly readvertise static routes, include the `readvertise` option. Include the `readvertise` option when configuring an individual route in the `route` portion of the static statement to override a `retain` option specified in the `defaults` portion of the statement.

```
[edit routing-options static (defaults | route)]
no-readvertise;
```

Configure a Default Route

To configure a default route, configure a next-hop address and include the `retain` option:

```
[edit]
user@host# set routing-options static route default next-hop address retain
[edit]
user@host# show
routing-options {
  static {
    route default {
      next-hop address;
      retain;
    }
  }
}
```

Propagate Static Routes into Routing Protocols

A common way to propagate static routes into the various routing protocols is to configure the routes so that the next-hop router is the loopback address (commonly, 127.0.0.1). Doing this with the JUNOS software (by including a statement such as `route address/mask-length next-hop 127.0.0.1`) does not propagate the static routes, because the forwarding table ignores static routes whose next-hop router is the loopback address. To propagate static routes into the routing protocols, configure them as follows:

```
[edit]
user@host# set routing-options static route destination-prefix discard
[edit]
user@host# show
routing-options {
  static {
    route destination-prefix {
      discard;
    }
  }
}
```

In this configuration, you use the `discard` option instead of `reject` because `discard` does not send an ICMP unreachable message for each packet that it drops.

Examples: Configure Static Routes

Configure a default route through the next-hop router 192.238.52.33:

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 192.238.52.33
[edit]
user@host# show
routing-options {
  static {
    route 0.0.0.0/0 next-hop 192.238.52.33;
  }
}
```

Configure static routes that are retained in the forwarding table when the routing software shuts down normally:

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 192.168.1.254 retain
[edit]
user@host# set routing-options static route 10.1.1.1/32 next-hop 127.0.0.1 retain
[edit]
user@host# show
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 192.168.1.254;
      retain;
    }
    route 10.1.1.1/32 {
      next-hop 127.0.0.1;
      retain;
    }
  }
}
```

Configure a static route and have it propagate into the routing protocols. In this example, do *not* specify the route as 143.172.0.0/6 next-hop 127.0.0.1.

```
[edit]
user@host# set routing-options static route 143.172.0.0/6 discard
[edit]
user@host# show
routing-options {
  static {
    route 143.172.0.0/6 discard;
  }
}
```

Configure Aggregate Routes

Route aggregation allows you to combine groups of routes with common addresses into a single entry in the routing table. This decreases the size of the routing table as well as the number of route advertisements sent by the router.

An aggregate route becomes active when it has one or more *contributing routes*. A contributing route is an active route that is a more specific match for the aggregate destination. For example, for the aggregate destination 128.100.0.0/16, routes to 128.100.192.0/19 and 128.100.67.0/24 are contributing routes, but routes to 128.0.0.0./8, 128.0.0.0/16, and 128.100.0.0/16 are not.

A route can contribute only to a single aggregate route. However, an active aggregate route can recursively contribute to a less specific matching aggregate route. For example, an aggregate route to the destination 128.100.0.0/16 can contribute to an aggregate route to 128.96.0.0/13.

When an aggregate route becomes active, it is installed in the routing table with the following information:

Reject next hop—This means that if a more specific packet does not match a more specific route, the packet is rejected and an ICMP unreachable message is sent to the packet's originator.

Metric value as configured with the aggregate statement.

Preference value that results from the policy filter on the primary contributor, if a filter was specified—Otherwise, the preference value as configured in the aggregate statement.

AS path as configured in the aggregate statement, if any—Otherwise, the path is computed by aggregating the paths of all contributing routes.

Community as configured in the aggregate statement, if any—Otherwise, the community is formed from a union of all communities from all contributing routes.



Note

You can configure only one aggregate route for each destination prefix.

To configure aggregate routes in the default routing table (inet.0), include the static statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  aggregate {
    defaults {
      aggregate-options;
    }
    route destination-prefix {
      policy policy-name;
      aggregate-options;
    }
  }
}
```

To configure aggregate routes in one of the other routing tables, or to explicitly configure aggregate routes in the default routing table (inet.0), include the aggregate statement at the [edit routing-options rib *routing-table*] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    aggregate {
      defaults {
        aggregate-options;
      }
      route destination-prefix {
        policy policy-name;
        aggregate-options;
      }
    }
  }
}
```

The aggregate statement consists of two parts:

defaults part—Here you specify global aggregate route options. These are treated as global defaults and apply to all the aggregate routes you configure in the aggregate statement. This part of the aggregate statement is optional.

route part—Here you configure individual aggregate routes. In this part of the aggregate statement, you optionally can configure aggregate route options. These options apply to the individual destination only and override any options you configured in the defaults part of the aggregate statement.

The following sections explain how to configure aggregate routes.

Specify the Destination of the Aggregate Route on page 102

Specify Aggregate Route Options on page 103

Specify Policy with Aggregate Routes on page 107

Advertise Aggregate Routes on page 107

Specify the Destination of the Aggregate Route

When you configure an individual aggregate route in the route part of the aggregate statement, specify the destination of the route (in route *destination-prefix*) in one of the following ways:

network/masklen, where *network* is the network portion of the IP address and *masklen* is the destination prefix length.

default if this is the default route to the destination. This is equivalent to specifying an IP address of 0.0.0.0/0.

Specify Aggregate Route Options

In the defaults and route parts of the aggregate statement, you can specify *aggregate-options*, which define additional information about aggregate routes that is included with the route when it is installed in the routing table. All aggregate options are optional. Aggregate options that you specify in the defaults part of the aggregate statement are treated as global defaults and apply to all the aggregate routes you configure in the aggregate statement. Aggregate options that you specify in the route part of the aggregate statement override any global aggregate options and apply to that destination only.

To configure aggregate route options, include one or more of them in the defaults or route part of the aggregate statement. Each of these options is explained in the sections that follow.

```
[edit]
routing-options {
  aggregate {
    (defaults | route) {
      (active | passive);
      as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
        <aggregator as-number in-address>;
      community ([community-ids] | no-advertise | no-export |
        no-export-subconfed | none);
      discard;
      (full | brief);
      (metric | metric2 | metric3 | metric4)metric <type type>;
      (preference | preference2 | color | color2) preference
        <type type>;
      (tag | tag2) string;
    }
  }
}
```

The following sections explain how to specify aggregate route options:

Specify the Route Metric on page 104

Specify the Route Preference on page 104

Specify a Next Hop for a Route on page 104

Specify Community Information on page 105

Specify the AS Path on page 105

Specify Which AS Numbers to Include in the Aggregate Route on page 106

Specify the OSPF Tag on page 106

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table on page 106

Specify the Route Metric

You can specify up to four metric values, starting with `metric` (for the first metric value) and continuing with `metric2`, `metric3`, and `metric4` by including one or more of the following statements at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
(metric | metric2 | metric3 | metric4) metric <type type>;
```

In the `type` option, you can specify the type of route.

Specify the Route Preference

By default, aggregate routes have a preference value of 130. If the routing table contains a dynamic route to a destination that has a better (lower) preference value than this, the dynamic route is chosen as the active route and is installed in the forwarding table.

To modify the default preference value, specify a primary preference value (`preference`). You also can specify secondary preference value (`preference2`); and colors, which are even finer-grained preference values (`color` and `color2`). To do this, include one or more of the following statements at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
(preference | preference2 | color | color2) preference
<type type>;
```

The preference value can be a number in the range 1 through 255, with a lower number indicating a more preferred route. For more information about preference values, see “Route Preferences” on page 5.

In the `type` option, you can specify the type of route.

Specify a Next Hop for a Route

By default, when aggregate routes are installed in the routing table, the next hop is configured as a reject route. That is, the packet is rejected and an ICMP unreachable message is sent to the packet’s originator.

When you configure an individual route in the `route` part of the aggregate statement, or when you configure the defaults for aggregate routes, you can specify a discard next hop. This means that if a more specific packet does not match a more specific route, the packet is rejected and a reject route for this destination is installed in the routing table, but ICMP unreachable messages are not sent. The discard next-hop feature allows you to originate a summary route, which is advertisable through dynamic routing protocols, and allows you to discard received traffic that does not match a more specific route than the summary route.

For example:

```
[edit routing-options aggregate defaults]
user@host#set discard
```

Specify Community Information

By default, no BGP community information is associated with aggregate routes. To associate community information with the routes, include the community option at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
  community ([ community-ids ] | no-advertise | no-export | no-export-subconfed | none);
```

community-ids is one or more community identifiers in the format *as-number:community-value*. *as-number* is the AS number and can be a value in the range 1 through 65534. *community-value* is the community identifier and can be a number in the range 0 through 65535.

You also can specify *community-ids* as one of the following well-known community names, which are defined in RFC 1997:

no-export—Routes containing this community name are not advertised outside a BGP confederation boundary.

no-advertise—Routes containing this community name are not advertised to other BGP peers.

no-export-subconfed—Routes containing this community name are not advertised to external BGP peers, including peers in other members' ASs inside a BGP confederation.

Specify the option **none** to explicitly not include BGP community information with an aggregate route. Include this option when configuring an individual route in the route portion of the aggregate statement to override a community option specified in the defaults portion of the statement.

Specify the AS Path

By default, the AS path for aggregate routes is built from the component routes. To manually specify the AS path and associate AS path information with the routes, include the *as-path* option at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
  as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
    <aggregator as-number in-address>
```

as-path is the AS path to include with the route. It can include a combination of individual AS path numbers and AS sets. Enclose sets in brackets ([]). The first AS number in the path represents the AS immediately adjacent to the local AS. Each subsequent number represents an AS that is progressively farther from the local AS, heading toward the origin of the path.

You also can specify the AS path using the BGP origin attribute, which indicates the origin of the AS path information:

igp—Path information originated within the local AS.

egp—Path information originated in another AS.

incomplete—Path information learned by some other means.

To attach the BGP ATOMIC_AGGREGATE path attribute to the aggregate route, specify the atomic-aggregate option. This path attribute indicates that the local system selected a less specific route rather than a more specific route.

To attach the BGP AGGREGATOR path attribute to the aggregate route, specify the aggregator option. When using this option, you must specify the last AS number that formed the aggregate route (encoded as 2 octets), followed by the IP address of the BGP system that formed the aggregate route.

Specify Which AS Numbers to Include in the Aggregate Route

By default, all AS numbers from all contributing paths are included in the aggregate route's path. To include only the longest common leading sequences from the contributing AS paths, include the brief option when configuring the route. If doing this results in AS numbers being omitted from the aggregate route, the BGP ATOMIC_ATTRIBUTE path attribute is included with the aggregate route.

```
[edit routing-options static (defaults | route)]
  brief;
```

To explicitly have all AS numbers from all contributing paths be included in the aggregate route's path, include the full option when configuring routes. Include this option when configuring an individual route in the route portion of the aggregate statement to override a retain option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
  full;
```

Specify the OSPF Tag

By default, no OSPF tag strings are associated with aggregate routes. You can specify up to two OSPF tag strings by including the tag and tag2 options at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
  (tag | tag2) string;
```

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table

By default, aggregate routes are removed from the routing and forwarding tables when they become inactive. To have an aggregate route remain continually installed in the routing and forwarding tables, include the passive option when configuring the route:

```
[edit routing-options static (defaults | route)]
  passive;
```

Routes that have been configured to remain continually installed in the routing and forwarding tables are marked with reject next hops when they are inactive.

To explicitly remove aggregate routes when they become inactive, include the `active` option when configuring routes. Include this option when configuring an individual route in the route portion of the aggregate statement to override a `retain` option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
active;
```

Specify Policy with Aggregate Routes

You can associate a routing policy when configuring an aggregate route's destination prefix in the routes part of the aggregate statement. Doing so provides the equivalent of an import routing policy filter for the destination prefix. That is, each potential contributor to an aggregate route, along with any aggregate options, is passed through the policy filter. The policy then can accept or reject the route as a contributor to the aggregate route and, if the contributor is accepted, the policy can modify the default preferences. The contributor with the numerically smallest prefix becomes the most preferred, or *primary*, contributor. A rejected contributor still can contribute to less specific aggregate route. If you do not specify a policy filter, all candidate routes contribute to an aggregate route.

To associate a routing policy with an aggregate route, include the policy statement when configuring the route:

```
[edit routing-options static (defaults | route)]
policy policy-name;
```

Advertise Aggregate Routes

After you have configured aggregate routes, you can have a protocol advertise the routes by configuring a policy that is then exported by a routing protocol. For example:

```
[edit]
policy-options {
  policy-statement advertise-aggregate-routes {
    term first-term {
      from protocol aggregate;
      then accept;
    }
    term second-term {
      then next-policy;
    }
  }
}
protocols {
  bgp {
    export advertise-aggregate-routes;
    ...
  }
}
```

Configure Generated Routes

Generated routes are used as the *route of last resort*. A packet is forwarded to the route of last resort when the routing tables have no information about how to reach that packet's destination. One use of route generation is to generate a default route to use if the routing table contains a route from a peer on a neighboring backbone.

A generated route becomes active when it has one or more *contributing routes*. A contributing route is an active route that is a more specific match for the generated destination. For example, for the destination 128.100.0.0/16, routes to 128.100.192.0/19 and 128.100.67.0/24 are contributing routes, but routes to 128.0.0.0/8, 128.0.0.0/16, and 128.100.0.0/16 are not.

A route can contribute only to a single generated route. However, an active generated route can recursively contribute to a less specific matching generated route. For example, a generated route to the destination 128.100.0.0/16 can contribute to a generated route to 128.96.0.0/13.

By default, when generated routes are installed in the routing table, the next hop is chosen from the primary contributing route.



Note

Currently, you can configure only one generated route for each destination prefix.

To configure generated routes in the default routing table (inet.0), include the generate statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  generate {
    defaults {
      generate-options;
    }
    route destination-prefix {
      policy policy-name;
      generate-options;
    }
  }
}
```

To configure aggregate routes in one of the other routing tables, or to explicitly configure aggregate routes in the default route table (inet.0), include the aggregate statement at the [edit routing-options rib *routing-table*] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    generate {
      defaults {
        generate-options;
      }
      route destination-prefix {
        policy policy-name;
        generate-options;
      }
    }
  }
}
```

The generate statement consists of two parts:

defaults—Here you specify global generated route options. These are treated as global defaults and apply to all the generated routes you configure in the generate statement. This part of the generate statement is optional.

route—Here you configure individual generated routes. In this part of the generate statement, you optionally can configure generated route options. These options apply to the individual destination only and override any options you configured in the defaults part of the generate statement.

The following sections explain how to configure generated routes.

Specify the Destination of a Generated Route on page 109

Specify Generated Route Options on page 110

Specify Policy with Generated Routes on page 114

Specify the Destination of a Generated Route

When you configure an individual generated route in the route part of the generate statement, specify the destination of the route (in route *destination-prefix*) in one of the following ways:

network/masklen, where *network* is the network portion of the IP address and *masklen* is the destination prefix length.

default if this is the default route to the destination. This is equivalent to specifying an IP address of 0.0.0.0/0.

Specify Generated Route Options

In the defaults and route parts of the generate statement, you can specify *generate-options*, which define additional information about generated routes that is included with the route when it is installed in the routing table. All generated options are optional. Generated options that you specify in the defaults part of the generate statement are treated as global defaults and apply to all the generated routes you configure in the generate statement. Generated options that you specify in the route part of the generate statement override any global aggregate options and apply to that destination only.

To configure generated route options, include one or more of them in the defaults or route part of the generate statement. Each of these options is explained in the sections that follow.

```
[edit]
routing-options {
  generate {
    (defaults | route) {
      (active | passive);
      as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
        <aggregator as-number in-address>;
      community ([community-ids] | no-advertise | no-export | no-export-subconfed | none);
      discard;
      (full | brief);
      (metric | metric2 | metric3 | metric4) metric <type type>;
      (preference | preference2 | color | color2) preference <type type>;
      (tag | tag2) string;
    }
  }
}
```

The following sections explain how to specify generated route options:

Specify the Route Metric on page 111

Specify the Route Preference on page 111

Specify a Next Hop for a Route on page 111

Specify Community Information on page 112

Specify the AS Path on page 112

Specify the OSPF Tag on page 113

Specify Which AS Numbers to Include in the Generated Route on page 113

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table on page 113

Specify the Route Metric

You can specify up to four metric values, starting with `metric` (for the first metric value) and continuing with `metric2`, `metric3`, and `metric4` by including one or more of the following statements at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
(metric | metric2 | metric3 | metric4) metric <type type>;
```

In the `type` option, you can specify the type of route.

Specify the Route Preference

By default, generated routes have a preference value of 130. If the JUNOS routing table contains a dynamic route to a destination that has a better (lower) preference value than this, the dynamic route is chosen as the active route and is installed in the forwarding table.

To modify the default preference value, specify a primary preference value (`preference`). You also can specify a secondary preference value (`preference2`); and colors, which are even finer-grained preference values (`color` and `color2`). To do this, include one or more of the following statements at the `[edit routing-options static (defaults | route)]` hierarchy level:

```
[edit routing-options static (defaults | route)]
(preference | preference2 | color | color2) preference <type type>;
```

The preference value can be a number in the range 1 through 255, with a lower number indicating a more preferred route. For more information about preference values, see “Route Preferences” on page 5.

In the `type` option, you can specify the type of route.

Specify a Next Hop for a Route

By default, when generated routes are installed in the routing table, the next hop is chosen from the primary contributing route.

When you configure an individual route in the `route` part of the `generate` statement, or when you configure the defaults for generated routes, you can specify a `discard` next hop. This means that if a more specific packet does not match a more specific route, the packet is rejected and a reject route for this destination is installed in the routing table, but ICMP unreachable messages are not sent. The `discard` next-hop feature allows you to originate a summary route, which is advertisable through dynamic routing protocols, and allows you to discard received traffic that does not match a more specific route than the summary route.

For example:

```
[edit routing-options generate route 1.0.0.0/8]
user@host# set discard
```

Specify Community Information

By default, no BGP community information is associated with generated routes. To associate community information with the routes, include the community option at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
  community ([ community-ids ] | no-advertise | no-export | no-export-subconfed | none);
```

community-ids is one or more community identifiers in the format *as-number:community-value*. *as-number* is the AS number and can be a value in the range 1 through 65534. *community-value* is the community identifier and can be a number in the range 0 through 65535.

You also can specify *community-ids* as one of the following well-known community names, which are defined in RFC 1997:

no-export—Routes containing this community name are not advertised outside a BGP confederation boundary.

no-advertise—Routes containing this community name are not advertised to other BGP peers.

no-export-subconfed—Routes containing this community name are not advertised to external BGP peers, including peers in other members' ASs inside a BGP confederation.

Specify the option **none** to explicitly exclude BGP community information with a generated route. Include this option when configuring an individual route in the route portion of the generate statement to override a community option specified in the defaults portion of the statement.

Specify the AS Path

By default, no AS path information is associated with generated routes. To associate AS path information with the routes, include the as-path option at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
  as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate>
    <aggregator as-number in-address>;
```

as-path is the AS path to include with the route. It can include a combination of individual AS path numbers and AS sets. Enclose sets in brackets ([]). The first AS number in the path represents the AS immediately adjacent to the local AS. Each subsequent number represents an AS that is progressively farther from the local AS, heading toward the origin of the path.

You also can specify the AS path using the BGP origin attribute, which indicates the origin of the AS path information:

igp—Path information originated within the local AS.

egp—Path information originated in another AS.

incomplete—Path information learned by some other means.

To attach the BGP ATOMIC_AGGREGATE path attribute to the generated route, specify the atomic-aggregate option. This path attribute indicates that the local system selected a less specific route rather than a more specific route.

To attach the BGP AGGREGATOR path attribute to the aggregate route, specify the aggregator option. When using this option, you must specify the last AS number that formed the generated route (encoded as 2 octets), followed by the IP address of the BGP system that formed the generated route.

Specify the OSPF Tag

By default, no OSPF tag strings are associated with generated routes. You can specify up to two OSPF tag strings by including the tag and tag2 options at the [edit routing-options static (defaults | route)] hierarchy level:

```
[edit routing-options static (defaults | route)]
(tag | tag2) string;
```

Specify Which AS Numbers to Include in the Generated Route

By default, all AS numbers from all contributing paths are included in the generated route's path. To include only the longest common leading sequences from the contributing AS paths, include the brief option when configuring the route. If doing this results in AS numbers being omitted from the generated route, the BGP ATOMIC_ATTRIBUTE path attribute is included with the generated route.

```
[edit routing-options static (defaults | route)]
brief;
```

To explicitly have all AS numbers from all contributing paths be included in the aggregate route's path, include the full option when configuring routes. Include this option when configuring an individual route in the route portion of the aggregate statement to override a retain option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
full;
```

Specify Whether Inactive Routes Are Removed from the Routing or Forwarding Table

By default, generated routes are removed from the routing and forwarding tables when they become inactive. To have a generated route remain continually installed in the routing and forwarding tables, include the passive option when configuring the route:

```
[edit routing-options static (defaults | route)]
passive;
```

Routes that have been configured to remain continually installed in the routing and forwarding tables are marked with reject next hops when they are inactive.

To explicitly remove generated routes when they become inactive, include the `active` option when configuring routes. Include this option when configuring an individual route in the route portion of the generate statement to override a retain option specified in the defaults portion of the statement.

```
[edit routing-options static (defaults | route)]
active;
```

Specify Policy with Generated Routes

You optionally can associate a routing policy when configuring a generated route's destination prefix in the routes part of the generate statement. Doing so provides the equivalent of an import routing policy filter for the destination prefix. That is, each potential contributor to a generated route, along with any generate options, is passed through the policy filter. The policy can accept or reject the route as a contributor to the generated route and, if the contributor is accepted, the policy can modify the default preferences. The contributor with the numerically smallest prefix becomes the most preferred, or *primary*, contributor. A rejected contributor still can contribute to less specific generated route. If you do not specify a policy filter, all candidate routes contribute to a generated route.

To associate a routing policy with an aggregate route, include the policy statement at the [edit routing-options static route] hierarchy level:

```
[edit routing-options static route]
policy policy-name;
```

Configure Martian Addresses

Martian addresses are host or network addresses about which all routing information is ignored. They commonly are sent by improperly configured systems on the network and have destination addresses that are obviously invalid.

In IPv4, the following are the default martian addresses:

```
0.0.0.0/8
127.0.0.0/8
128.0.0.0/16
191.255.0.0/16
192.0.0.0/24
223.255.255.0/24
240.0.0.0/4
```

Add Martian Addresses

To add martian addresses to the list of default martian addresses in the default routing table (inet.0), include the martians statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  martians {
    destination-prefix match-type;
  }
}
```

To add martian addresses to the list of default martian addresses in any other routing tables, or to explicitly add martian addresses to the list of default martian addresses in the default routing table (inet.0), include the martians statement at the [edit routing-options rib *rib-table*] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    martians {
      destination-prefix match-type;
    }
  }
}
```

In *destination-prefix*, specify the routing destination in one of the following ways:

network/mask-length—*network* is the network portion of the IP address and *mask-length* is the destination prefix length.

default—If this is the default route to the destination. This is equivalent to specifying the IP address 0.0.0.0/0.

In *match-type*, specify the type of match to apply to the destination prefix. It can be one of the operations listed in Table 6 on page 58.

Remove Martian Addresses

To delete a martian address from within a range of martian addresses, include the *allow* option in the martians statement. This option removes an exact prefix that is within a range of addresses that has been specified to be martian addresses.

To delete a martian address from the default routing table (inet.0), include the martians statement at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
  martians {
    destination-prefix match-type allow;
  }
}
```

To delete a martian address from any other routing tables, or to explicitly delete a martian address from the default routing table (inet.0), include the martians statement at the [edit routing-options rib *rib-table*] hierarchy level:

```
[edit]
routing-options {
  rib routing-table {
    martians {
      destination-prefix match-type allow;
    }
  }
}
```