

## Chapter 3

# Routing Policy Configuration Statements

To create a routing policy, you can include the `policy-options` statement in the configuration:

```
[edit]
policy-options {
  as-path name regular-expression;
  as-path-group group-name;
  community name {
    invert-match;
    members [ community-ids ];
  }
  damping name {
    disable;
    half-life minutes;
    max-suppress minutes;
    reuse number;
    suppress number;
  }
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        policy subroutine-policy-name;
        prefix-list name;
        route-filter destination-prefix match-type <actions>;
        source-address-filter destination-prefix match-type <actions>;
      }
      to {
        match-conditions;
        policy subroutine-policy-name;
      }
      then actions;
      default-action (accept | reject);
    }
  }
  prefix-list name {
    ip-addresses;
  }
}
```

```

protocols {
  protocol-name {
    import [ policy-names ];
    export [ policy-names ];
  }
}

```

This section includes the following minimum configurations:

Minimum Routing Policy Configuration on page 38

Minimum Routing Policy Chain Configuration on page 39

Minimum Subroutine Configuration on page 40

## Minimum Routing Policy Configuration

---

To define and apply a routing policy, you must include at least the following statements at the [edit policy-options] and [edit protocols] hierarchy levels. At the [edit protocols] hierarchy level, you can define one or more policy names.

```

[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        prefix-list name;
        route-filter destination-prefix match-type <actions>;
        source-address-filter destination-prefix match-type <actions>;
      }
      to {
        match-conditions;
      }
      then actions;
    }
    prefix-list name {
      ip-addresses;
    }
  }
}
protocols {
  protocol-name {
    import [ policy-names ];
    export [ policy-names ];
  }
}

```

## Minimum Routing Policy Chain Configuration

---

To define and apply a routing policy chain, you must include at least the following statements at the [edit policy-options] and [edit protocols] hierarchy levels. At the [edit protocols] hierarchy level, you can define a chain of policy names that are evaluated in order.

```
[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        prefix-list name;
        route-filter destination-prefix match-type <actions>;
        source-address-filter destination-prefix match-type <actions>;
      }
      to {
        match-conditions;
      }
      then actions;
    }
  }
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        prefix-list name;
        route-filter destination-prefix match-type <actions>;
        source-address-filter destination-prefix match-type <actions>;
      }
      to {
        match-conditions;
      }
      then actions;
    }
  }
  prefix-list name {
    ip-addresses;
  }
}
protocols {
  protocol-name {
    import [ policy-names ];
    export [ policy-names ];
  }
}
```

## Minimum Subroutine Configuration

---

To configure a routing policy that calls a subroutine from another routing policy, you must include at least the following statements at the [edit policy-options] and [edit protocols] hierarchy levels. At the [edit protocols] hierarchy level, you can define one or more policy names.

```
[edit]
policy-options {
  policy-statement subroutine-policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        prefix-list name;
        route-filter destination-prefix match-type <actions>;
        source-address-filter destination-prefix match-type <actions>;
      }
      to {
        match-conditions;
      }
      then actions;
    }
  }
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        policy subroutine-policy-name;
      }
      to {
        policy subroutine-policy-name;
      }
      then actions;
    }
  }
}
protocols {
  protocol-name {
    import [ policy-names ];
    export [ policy-names ];
  }
}
```