

Chapter 14

Configure Routing Instances

A routing instance is a routing entity for a router. You can configure multiple routing instances of OSPF. Each instance contains a routing table, applied routing policies, routing table group, interfaces that belong to that instance, and an OSPF configuration related to that instance.

You configure a primary routing instance at the [edit protocols] hierarchy level. You configure additional routing instances at the [edit routing-instances] hierarchy level.

You use routing instances to:

- Create administrative separation in a large network to segregate customer traffic and associated settings. The customers see only the routes belonging to them.

- Create overlay networks in which separate services are routed only towards routers participating in that service, such as voice. The overlay network isolates routes belonging to one service from another service by exporting routes, applying tags, and filtering based on tags.

This chapter includes the following topics:

- Routing Instances Statement Hierarchy on page 160

- Routing Instances Minimum Configuration on page 162

- Configure Multiple Routing Instances on page 163

- Summary of Multiple Routing Instances Statements on page 167

Routing Instances Statement Hierarchy

To configure routing instances, include the following statements at the [edit routing-instances] and [edit routing-options] hierarchy levels of the configuration:

```
[edit]
routing-instances {
  routing-instance-name {
    interface interface-name {
      protocols {
        ospf {
          ospf-configuration;
        }
      }
    }
  }
  routing-options {
    aggregate {
      defaults {
        aggregate-options;
      }
    }
    route destination-prefix {
      policy policy-name;
      aggregate-options;
    }
  }
  autonomous-system autonomous-system <loops number>;
  confederation confederation-autonomous-system members autonomous-system;
  fate-sharing {
    cost value;
    from address [to address];
  }
  forwarding-table {
    export [ policy-names ];
  }
  generate {
    defaults {
      generate-options;
    }
    route destination-prefix {
      policy policy-name;
      generate-options;
    }
  }
  interface-routes {
    rib-group routing-table-name;
  }
  martians {
    destination-prefix match-type <allow>;
  }
  multicast {
    scope scope-name {
      interface [ interface-name ];
      prefix destination-prefix;
    }
  }
  options {
    syslog (level level | upto level);
  }
  rib routing-table {
    static {
      defaults {
        static-options;
      }
    }
  }
}
```

```

        route destination-prefix {
            next-hop;
            static-options;
        }
    }
}
aggregate {
    defaults {
        aggregate-options;
    }
    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>;
}
}
rib-group {
    group-name {
        import-rib [ routing-table-name ];
        export-rib routing-table-name;
    }
}
route-record;
router-id address;
{
    defaults {
        static-options;
    }
    route destination-prefix {
        next-hop;
        static-options;
    }
}
traceoptions {
    file name <replace> <size size> <files number> <no-stamp> <world-readable>;
    flag flag <flag-modifier> <disable>;
}
}

```

For more detailed information, see “[edit routing-options] Hierarchy Level” on page 20 and “[edit protocols] Hierarchy Level” on page 10.

Routing Instances Minimum Configuration

To configure routing instances for OSPF, you must include at least the following statements in the configuration:

```
[edit]
protocols {
  ospf {
    rib-group common;
    area area-id {
      interface all;
    }
  }
}
routing-instances {
  routing-instance-name {
    interface interface-name;
    protocols {
      ospf {
        rib-group common;
        area area-id {
          interface all;
        }
      }
    }
  }
}
routing-options {
  rib-groups {
    common {
      import-rib [inet.0 routing-instance-name.inet.0];
    }
  }
}
policy-statement apply-tag {
  term instance-name1 {
    from instance routing-instance-name1;
    then {
      tag string1;
      accept;
    }
  }
  term routing-instance-name2 {
    from instance routing-instance-name2;
    then {
      tag string2;
      accept;
    }
  }
}
policy-statement filter-on-routing-instance-name1 {
  from protocol ospf;
  tag string1;
}
then accept;
}
}
policy-statement filter-on-routing-instance-name2 {
  from protocol ospf;
  tag string2;
}
then accept;
}
}
```



You can configure only one routing instance per logical interface.

Configure Multiple Routing Instances

To configure multiple routing instances, perform the following tasks:

Configure the OSPF primary instance at the [edit protocols ospf] hierarchy level with the statements needed for your network so that routes are installed in inet.0 and therefore are installed in the forwarding table. Make sure to include the routing table group. For more information, see “Configure the Primary OSPF Instance” on page 167 and “Configure a Routing Table Group” on page 166.

Configure an OSPF routing instance for each additional OSPF routing entity, configuring the following:

Interfaces

Routing options

OSPF protocol statements belonging to that entity

Routing table group

For more information, see “Configure the Routing Instances” on page 165.

Configure a routing table group to install routes from the routing instance into the inet.0 routing table. You can do this in two ways:

1. Create a common routing table group so that:

Routes from the routing instances are installed in inet.0 and therefore installed in the forwarding table.

Routes from one router in a routing instance are forwarded to another router in the same routing instance.

2. Create a routing table group with just the routing table from one instance and inet.0 to keep the routes from going to other instances.

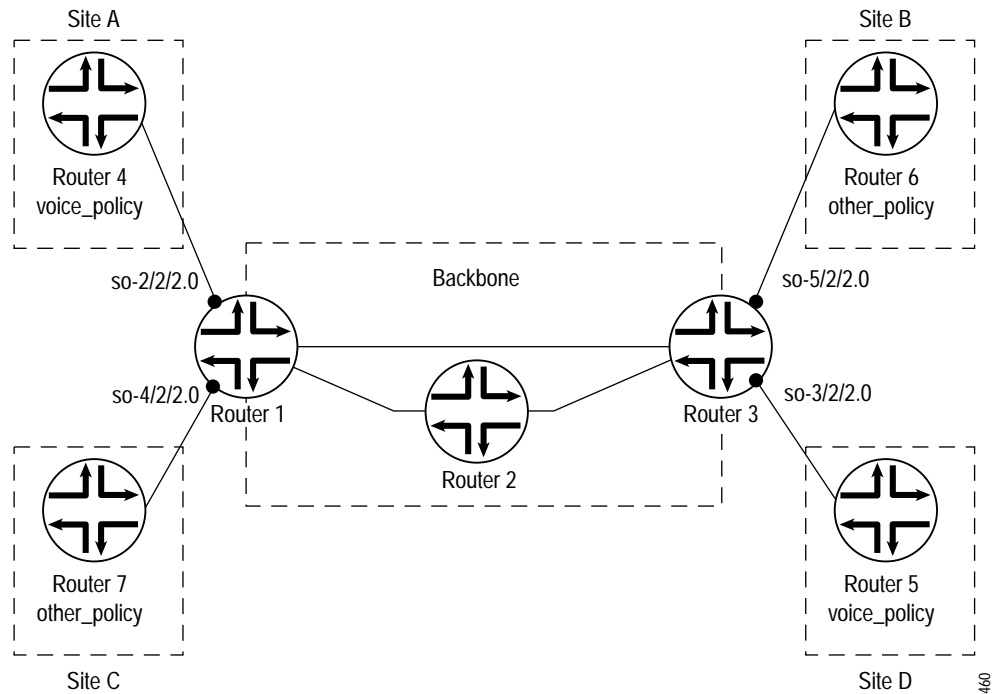
For more information, see “Configure a Routing Table Group” on page 166.

Create an export policy to export routes with a specific tag and to use that tag to export routes back into the instances. For more information, see “Define Routing Policies” on page 35 and “Configure a Routing Policy Statement” on page 166.

Example: Create Multiple Routing Instances

Figure 7 shows how you can use multiple routing instances of OSPF to segregate traffic within a large network. The network consists of three administrative entities: `voice_policy`, `other_policy`, and the backbone or core. Each entity is composed of several geographically separate sites that are connected by the backbone and managed by the backbone entity.

Figure 7: Configuration for Multiple Routing Instances



Sites A and D belong to the `voice_policy` routing instance. Sites B and C belong to the `other_policy` instance. Router 1 and Router 3 at the edge of the backbone connect the routing instances. Each run a separate OSPF instance (one per entity).

Router 1 runs three OSPF instances: one each for Site A (`voice_policy`), Site D (`other_policy`), and the backbone otherwise known as the primary instance. Router 3 also runs three OSPF instances; one each for Site B (`other_policy`), Site C (`voice_policy`), and the backbone (primary instance).

OSPF routes learned in one `voice_policy` site are distributed across the backbone instance to the other `voice_policy` site. Additionally, OSPF routes learned in one `other_policy` site are distributed across the core to the other `other_policy` site.

The backbone OSPF instance exports OSPF routes learned in the other instances and installs them in the backbone instances routing table as external routes tagged with their respective entity identifiers. The export policies select a routing instance from a route source and set the OSPF tag to a unique number corresponding to the routing instance entity.

External routes learned in the backbone OSPF instance are installed in routing tables of the three instances. Tags on these routes identify the entity to which they belong. The OSPF instance for to the `voice_policy` entity then exports these routes (learned from the backbone but also installed in the `voice_policy` routing table) into the `voice_policy` site if the tag on a route is appropriate (the same as the `voice_policy` entity identifier). Similarly, the OSPF instance corresponding to the `other_policy` entity exports routes tagged as `other_policy` entity routes into the `other_policy` entity. Export policies select OSPF external route tags to export them to the correct entity OSPF instance.

Configure Router 1

The sections that follow describe how to configure Router 1 in the backbone entity with multiple routing instances.

Configure the Routing Instances

Configure the routing instances for `voice_policy` and `other_policy`. Use all routes learned from the routing tables in the routing table group `common`. Export routes tagged as belonging to the routing instance:

```
[edit]
routing-instances {
  voice_policy {
    interface so-2/2/2.0;
    protocols {
      ospf {
        rib-group common;
        export filter-on-voice-policy;
        area 0.0.0.0 {
          interface all;
        }
      }
    }
  }
  routing-options;

  other_policy {
    interface so-4/2/2.0;
    protocols {
      ospf {
        rib-group common;
        export filter-on-other-policy;
        area 0.0.0.0 {
          interface all;
        }
      }
    }
  }
  routing-options {...};
}
```

Configure a Routing Table Group

Configure the routing table group common to share routes with inet.0 (in the backbone entity), the voice_policy.inet.0, and the other_policy.inet.0 routing tables:

```
[edit]
routing-options {
  rib-groups {
    common {
      import-rib [inet.0 voice_policy.inet.0 other_policy.inet.0];
    }
  }
}
```

Configure a Routing Policy Statement

Configure an export policy statement that a routing instance uses to export and tag OSPF routes learned from the routing instances:

```
[edit]
policy-options {
  policy-statement apply-tag {
    term voice_policy {
      from instance voice_policy;
      then {
        tag 10;
        accept;
      }
    }
  }
  term other_policy {
    from instance other_policy;
    then {
      tag 12;
      accept;
    }
  }
}

policy-statement filter-on voice_policy {
  from protocol ospf;
  tag 10;
}
then accept;
}

policy-statement filter-on other_policy {
  from protocol ospf;
  tag 12;
}
then accept;
}
```

Configure the Primary OSPF Instance

Configure the primary OSPF instance so that the routes learned from the routing instances are installed in inet.0 and to export the tagged routes from voice_policy and other_policy:

```
[edit]
protocols {
  ospf {
    rib-group common;
    export filter-on-voice_policy;
    export filter-on-other_policy;
    area 0.0.0.0 {
      interface so-2/2/2.0;
      interface so-4/2/2.0;
      passive;
    }
  }
}
```

Configure Router 3

The configuration for Router 3 is the same as for Router 1 except that the interface names might differ. In this example, the interface so-5/2/2.0 belongs to other_policy and so-3/2/2.0 belongs to voice_policy.

Summary of Multiple Routing Instances Statements

The following sections explain each of the routing instances configuration statements. The statements are organized alphabetically.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Enable a logical interface for a routing instance. You must include at least one interface statement in the configuration to enable the routing instance.
Options	<i>interface-name</i> —Name of the logical interface. For details about specifying interfaces, see the <i>JUNOS Internet Software Configuration Guide: Interfaces and Chassis</i> .
Usage Guidelines	See “Routing Instances Minimum Configuration” on page 162.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

routing-instances

Syntax	routing-instances { ... }
Hierarchy Level	[edit]

