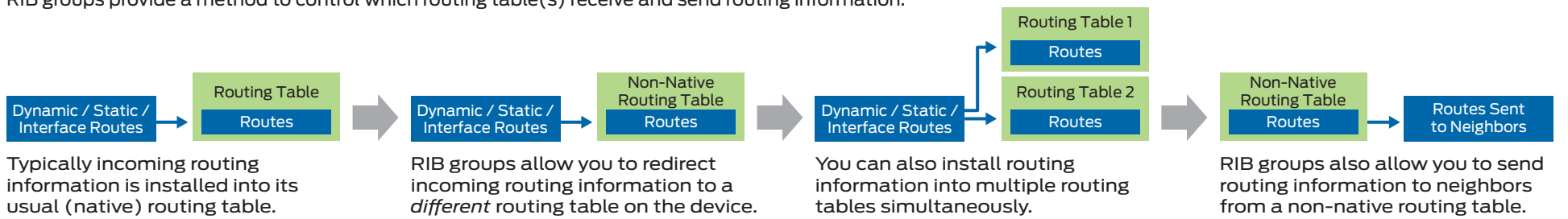# Using RIB Groups for Route Sharing in the Junos® OS

RIB Group Overview

RIB groups provide a method to control which routing table(s) receive and send routing information.

Typically incoming routing information is installed into its usual (native) routing table.

RIB groups allow you to redirect incoming routing information to a *different* routing table on the device.

You can also install routing information into multiple routing tables simultaneously.

RIB groups also allow you to send routing information to neighbors from a non-native routing table.

## Routing Instances and Tables

The process of populating a routing table is often thought of in simple terms — routes received from neighbors are placed into the appropriate routing table, and those routes are then sent on to other neighbors. However, there are more elements to this process.
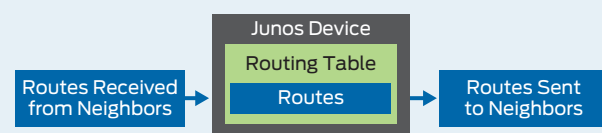
When routes arrive at the device, they are added to their related protocol RIB. The protocol RIB runs its algorithm to determine which routes should be installed into the related routing table, and the appropriate routes are then 'imported' into the table, say inet.0. To send routing information to neighbors, relevant routes are 'exported' from the routing table to the protocol RIB, and then sent on to neighbors.
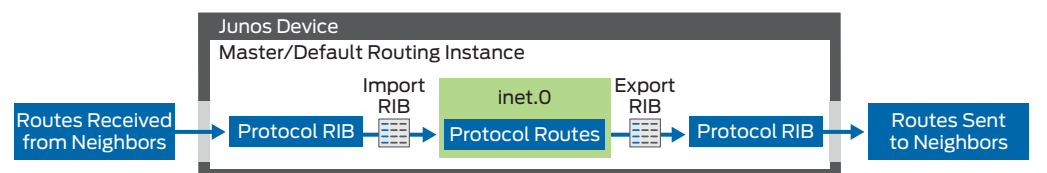
The sequence of steps described above happens per routing instance. By default, a Junos device has a single 'master' routing instance, thus routes received on an interface belonging to the master routing instance are installed into the related routing table in the master routing instance, say inet.0. This is in contrast to routes received on an interface belonging to another routing instance, say TEST. These routes are installed into the related routing table in the TEST routing instance, say TEST.inet.0.
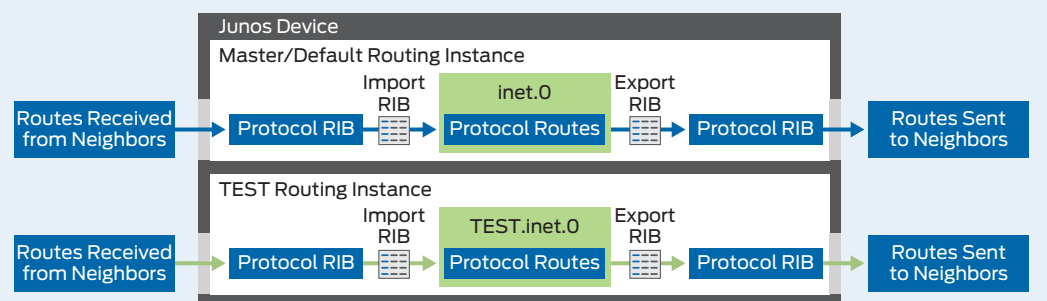
Configuration

Configuring RIB groups involves two steps.

1. Define a RIB group:

Table containing the desired routing information
Table(s) where the routing information should be added

```
routing-options {
   rib-groups {
      rg-name {
         import-rib [ src-table dest-table(s) ];
         import-policy [ policy-name ];
         export-rib table;
      }
   }
}
```

(Optional) Filter routes imported into the destination table(s)

(Optional) Non-default table from where routes are sent

2. Apply the RIB group (generalized):

```
protocols {
   protocol {
      rib-group rg-name;
   }
}
```

Route Sharing: Why Order Matters

The order of the `import-rib` statement defines the 'source' and 'destination' routing tables for route sharing. The source table is where the given routes are installed by default; the destination table(s) identifies where to share the routes.

For example, the statement `import-rib [ inet.0 TEST.inet.0 ]` does this:
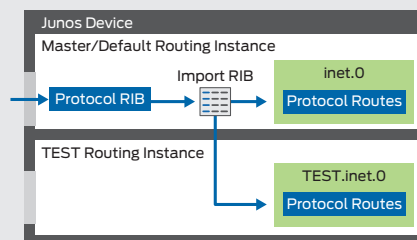
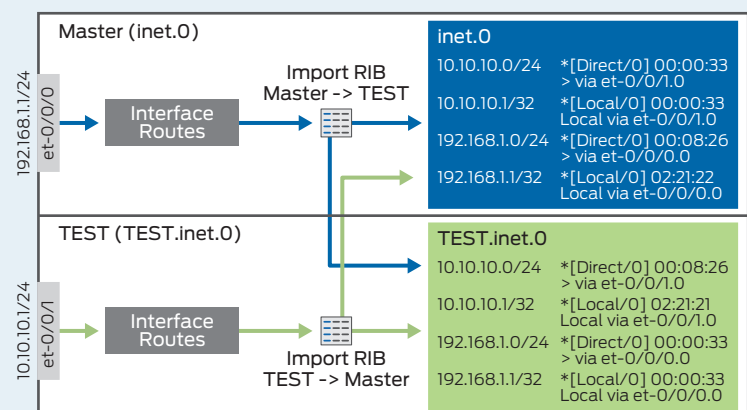Whereas, the statement `import-rib [ TEST.inet.0 inet.0 ]` does this:
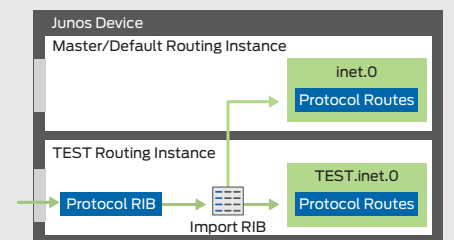
Example / Use Case

This example illustrates how to share directly connected (interface) routes. The Master and TEST routing instances each have a configured interface. The goal is to make each interface's local route information available in the other instance's routing table.

```
user@device# show interfaces
et-0/0/0 {  ## belongs to Master RI
   unit 0 {
      family inet {
         address 192.168.1.1/24;
      }
   }
}
et-0/0/1 {  ## assigned to TEST RI (at right)
   unit 0 {
      family inet {
         address 10.10.10.1/24;
      }
   }
}

user@device# show routing-options
interface-routes {
   rib-group inet to-TEST;  ## send Master int-routes to RG
}
rib-groups {
   to-TEST {  ## share Master int-routes with TEST
      import-rib [ inet.0 TEST.inet.0 ];
   }
   to-Master {  ## share TEST int-routes with Master
      import-rib [ TEST.inet.0 inet.0 ];
   }
}
```

```
user@device# show routing-instances
TEST {
   ...
   interface et-0/0/1.0;
   routing-options {
      interface-routes {
         rib-group inet to-Master;  ## send TEST int-routes to RG
      }
   }
}
```


For more information, visit: