Chapter 31
Route Importation

31.1 Route Importation Overview

`import` statements control the importation of routes from routing protocols and the installation of the routes in GateD's Routing Information Base (RIB). The format of an **`import`** statement varies depending on the source protocol. A given import statement applies to all routes that match the specified protocol and any other item specified on the import statement, such as **`tag`**, **`as`**, **`interface`**, **`gateway`**, and so on.

31.1.1 Route Filters

All the formats allow route filters as shown below. See "Chapter 28 Route Filtering" on page 129 for a detailed explanation of how they work. When no route filtering is specified (i.e., when **`restrict`** is specified on the first line of a statement), all routes from the specified source with any specified **`as`**, **`tag`**, and so on, will match that statement. If any filters are specified, only routes that also match the specified filters will be imported. Put differently, if any filters are specified, an **`all restrict`** is assumed at the end of the list.

31.1.2 Importing Routes into Different RIBS

Normally, routes from unicast routing protocols are only imported into the unicast RIB. Routes from multicast routing protocols are only imported into the multicast RIB. However, some multicast routing protocols do not maintain their own routing table, but rely on a unicast routing protocol instead. To support these protocols, unicast routes must be imported into the multicast RIB. If the routes are not imported, only interface routes will be available to those multicast protocols.

Because MPBGP is able to tag routes to indicate to which RIBs they apply, no additional configuration is required for BGP routes.

The RIP and Redirect protocols, however, do not tag routes as being for a specific Subsequent Address Family Indicator (SAFI). Hence, GateD must be configured to import RIP or Redirect routes into the multicast RIB. See "Examples of Importation into Multicast RIBs" on page 142 to see the exact syntax of the import protocol statement. One or more RIB names may be specified (where **`multicast`** and **`unicast`** appear) as in the example below:

```
import proto rip {
    all;
    198.0.0.0 masklen 8 refines multicast unicast;
};
```

This example keeps the normal behavior of allowing all RIP routes in the unicast RIB, but also imports all routes falling under 198/8 into the multicast RIB. Additional examples are included in "Examples of Importation into Multicast RIBs" on page 142.

To import OSPF routes into the multicast RIB, you currently must import all OSPF routes as follows:

```
ospf yes {
    defaults {
    ribs unicast multicast;
...
    };
...
};
```

You cannot import OSPF routes into only the multicast RIB. Attempting to do so will be flagged as a configuration error.

31.1.3 Import Inheritance

The following parameters can be specified at multiple places within a given **import** statement:

- **fromribs**
- **toribs**
- **restrict**
- **preference**

In the case of **restrict**, placing it on an **import proto** statement restricts importation of all routes that match the **import proto** statement. Including it on a *route_filter* restricts importation of all routes that match that route filter.

In the case of **preference**, **toribs**, and **fromribs**, the most specific instances of these values are assigned to a route. Thus, any such value specified on an **import proto** statement is used only if the *route_filter* that matches a route does not specify a value. The value specified on the most specific matching *route_filter* is used. If neither the **import proto** statement nor the matching *route_filter* specifies a value, then the default value for the protocol is used.

31.2 Route Importation Syntax

```
import proto bgp
    ( as ASN ) | ( aspath aspath-regular-expression
      origin ( any | igp | egp | incomplete ) )
    [ comm { communities_list } ]
    [ ext-comm { extended_communities_list } ]
    [ preference preference ]
    [ fromribs riblist ]
    [ [ toribs ] riblist ]
    { [ route_filter
      [ restrict |
      ( [ preference preference ]
        [ fromribs riblist ]
        [ [ toribs ] riblist ] ) ] ; ]
    };
import proto bgp
    ( as ASN ) | ( aspath aspath-regular-expression
      origin ( any | igp | egp | incomplete ) )
    [ comm { communities_list } ]
    [ ext-comm { extended_communities_list } ]
    restrict;

import proto ospfase
    [ tag tagvalue ]
    [ preference preference ]
    { [ route_filter
      [ restrict |
      ( [ preference preference ]
        [ [ toribs ] riblist ] ) ] ; ]
    };
import proto ospfase
    [ tag tagvalue ] restrict ;

import proto rip
   [ tag tagvalue | interface interface_list  |  gateway gateway_list ]
    [ preference preference ] [ [ toribs ] riblist ]
    { [ route_filter
      [ restrict |
      ( [ preference preference ]
        [ [ toribs ] riblist ] ) ] ; ]
    };
import proto rip
   [ tag tagvalue | interface interface_list  |  gateway gateway_list ]
    restrict ;

import proto ripng
   [ tag tagvalue | interface interface_list  |  gateway gateway_list ]
```

```
        [ preference preference ] [ [ toribs ] riblist ]
        { [ route_filter
          [ restrict |
          ( [ preference preference ]
            [ [ toribs ] riblist ] ) ] ; ]
        };
    import proto ripng
        [ tag tagvalue | interface interface_list | gateway gateway_list ]
        restrict ;

    import proto redirect
        [ interface interface_list | gateway gateway_list ]
        [ preference preference ] [ [ toribs ] riblist ]
        { [ route_filter
          [ restrict |
          ( [ preference preference ]
            [ [ toribs ] riblist ] ) ] ; ]
        };
    import proto redirect
        [ interface interface_list | gateway gateway_list ] restrict ;
```

route_filter:

route_filter specifies a list of matching filters and the corresponding action for each filter. For more information see “Chapter 28 Route Filtering” on page 129 in *Configuring GateD*.

communities_list and *extended_communities_list*:

communities_list specifies the set of communities that are to be matched. *extended_communities_list* specifies the set of extended communities to be matched. For more information, see “Chapter 30 BGP Communities” on page 133 in *Configuring GateD*.

31.3 Route Importation Defaults

```
    import proto bgp aspath “(.*)” origin any
        { all ; } ;
    import proto ospfase
        { all ; } ;
    import proto rip
        { all ; } ;
    import proto ripng
```

```
        { inet6 all ; } ;
    import proto redirect
        { all ; } ;
```

31.4 Route Importation Examples

Example 1

The following example will import only routes from AS 203 that are stamped with community 99:

```
import proto bgp as 203
    comm {
        comm-split 203 99
    }
{
    all;
};
```

Example 2

The following example will import only routes that do not originate from AS 690:

```
import proto bgp aspath “(.* 690)” origin any {
    all restrict;
};
import proto bgp aspath “(.*)” origin any {
    all;
};
```

Example 3

The following example will import all IPv4 routes from AS 200:

```
import proto bgp as 200 {
    inet all ;
} ;
```

Example 4

The following example will import all IPv6 routes from AS 65000:

```
import proto bgp as 65000 {
    inet6 all ;

} ;
```

Example 5

The following example will import all IPv6 and IPv4 routes from AS 65000:

```
import proto bgp as 65000 {
    all ;
} ;
```

Example 6

The following example will import all RIP routes:

```
import proto rip {
    all ;
} ;
```

Example 7

The following example will import RIP routes based on tags:

```
import proto rip tag 230 {
    all ;
} ;
```

31.4.1 Examples of Importation into Multicast RIBs

Example 1

Example 1 keeps the normal behavior of allowing all RIP routes in the unicast RIB, but also imports all routes falling under 198/8 into the multicast RIB.

```
import proto rip {

    all;

    198.0.0.0 masklen 8 refines multicast unicast;

};
```

Example 2

Example 2 imports all of the RIP routes into the multicast RIB (as well as the usual unicast RIB).

```
import proto rip {

    all multicast unicast;

};
```

Example 3

Example 3 imports all of the RIP routes into the unicast and multicast RIBs.

```
import proto rip unicast multicast {

    all ;

};
```

Example 4

Example 4 inserts all of the OSPF routes into the unicast and multicast RIBs.

```
    ospf yes {
        defaults {
            ribs unicast multicast;
    ...
        };
    ...
    };
```