

Chapter 27

Route Importation

fromribs

Name

fromribs - specifies the RIBs from which a route will be imported

Syntax

```
import proto ( ... import_parameters ... ) fromribs riblist
    route_filter fromribs riblist ;
```

Parameters

route_filter - a set of route filters specifying routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See “Chapter 28 Route Filtering” on page 129 of *Configuring GateD* for more information.

riblist - one of the following:

unicast - specifies that the routes in the unicast RIB are to be imported

multicast - specifies that the routes in the multicast RIB are to be imported

unicast multicast - specifies that the routes in both the unicast and multicast RIBs are to be imported

Description

fromribs is used to indicate the RIBs from which a route will be imported. It is currently only applicable to BGP because BGP can learn routes in either the unicast or multicast RIBs. When used on a filter, only routes in the specified RIB(s) are matched.

If just one of the unicast or multicast RIBs is specified for **fromribs** and a route to be imported is in both RIBs, the route will be imported to just the specified **fromribs**. This action can be overridden by a **toribs** specification.

Defaults

If **fromribs** is not specified, routes from either RIB are accepted and imported according to **toribs**. If **toribs** is not specified, the route is imported into the RIB, or RIBs from which it came.

Context

```
import statement  
route_filter
```

Examples

```
import proto bgp as 65534 {  
    all fromribs multicast ;  
} ;
```

This example only imports multicast routes learned from this AS.

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

import proto bgp

Name

`import proto bgp` - specifies BGP import policy

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 restrict
  ( as ASN ) | ( aspath aspath-regular-expression
    origin ( any | igp | egp | incomplete ) )
  [ comm { communities_list } ]
  [ ext-comm { extended_communities_list } ]
  restrict;
```

Parameters

as *ASN* - specifies that this import statement applies to routes learned from peers in AS *ASN*

comm { *communities_list* } - specifies that this import statement applies to routes learned with the communities specified in *communities_list*

ext-comm { *extended_communities_list* } - specifies that this import statement applies to routes learned with the extended communities specified in *extended_communities_list*

toribs - specifies RIBs into which routes are to be imported

fromribs - specifies RIBs from which routes are to be imported

preference *preference* - specifies the preference value to be assigned to routes that match this import statement or *route_filter*

route_filter - a set of route filters specifying particular routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

restrict - specifies that routes matching this import statement or *route_filter* are not to be considered in the route selection process

aspath *aspath-regular-expression* - specifies that this import statement or *route_filter* applies only to routes with an AS_PATH path attribute that matches *aspath-regular-expression*

origin any - specifies that this import statement applies to routes regardless of their ORIGIN path attribute

origin igp - specifies that this import statement applies to routes with an ORIGIN path attribute of IGP

origin egp - specifies that this import statement applies to routes with an ORIGIN path attribute of EGP

origin incomplete - specifies that this import statement applies to routes with an ORIGIN path attribute of incomplete

Description

Import statements specify the policy applied to routes being learned from a particular protocol. They specify what attributes are to be associated with routes when they are placed in the routing table, and the priority the routes are to be given in the route selection process.

BGP supports propagation control by the use of an AS path regular expression, which is documented in "Chapter 29 Matching AS Paths" on page 131 in *Configuring GateD*.

The **comm** option allows the specification of import policy based on the communities attributes found in the BGP update. If multiple communities are specified in the **comm** option, only updates carrying all of the specified communities will be matched. The **ext-comm** option allows the specification of import policy based on the communities attributes found in the BGP update. If multiple communities are specified in the **ext-comm** option, only updates carrying all of the specified communities will be matched. See "Chapter 30 BGP Communities" on page 133 in *Configuring GateD* for more information about communities.

Note that it is quite possible for several BGP import clauses to match a given update. If more than one clause matches, the first matching clause will be used. All later matching clauses will be ignored. For this reason, it is generally desirable to order import clauses from most to least specific. An import clause without a **comm** or **ext-comm** option will match any update regardless of the presence or absence of communities.

BGP stores any routes that were rejected implicitly by not being mentioned in a route filter, or explicitly with the **restrict** keyword, in the routing table with a negative preference. A negative preference prevents a route from becoming active, which prevents it from being installed in the forwarding table or exported to other protocols. This restriction mechanism alleviates the need to terminate and re-establish a session upon reconfiguration if importation policy is changed.

Defaults

By default, all routes from active protocols are imported with default preference.

Context

control statements

Examples

Example 1

The following example imports all routes learned from BGP peers in as 65534.

```
import proto bgp as 65534 {  
    all;  
};
```

Example 2

The following example imports all routes learned from BGP peers regardless of their AS:

```
import proto bgp aspath "(.*)" {  
    all;  
};
```

Example 3

The following example imports all routes learned from BGP peers with a community value of 65534 1000:

```
import proto bgp aspath "(.*)" comm { comm-split 65534 1000 } {  
    all;  
};
```

See Also

"Chapter 30 BGP Communities" on page 133 in *Configuring GateD*

"Chapter 31 Route Importation" on page 137 in *Configuring GateD*

"Communities" on page 77 in *Configuring GateD*

bgp on page 232

preference on page 618

import proto ospfase

Name

`import proto ospfase` - specifies OSPF import policy

Syntax

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

Parameters

route_filter - a set of route filters specifying particular routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

`tag tagvalue` - specifies that this import statement applies to routes learned with the tag *tagvalue*

`preference preference` - specifies the preference value to be assigned to routes that match this import statement or *route_filter*

`restrict` - specifies that routes which match this import statement or *route_filter* are not to be considered during the route selection process

Description

Import statements specify import policy. Import policy determines which, and in what manner, routes are to be added to GateDs internal routing table.

Due to the nature of OSPF, only the importation of ASE routes may be controlled. OSPF intra- and inter-area routes are always imported into the GateD routing table. If a tag is specified, the `import` clause will only apply to routes with the specified tag.

It is only possible to restrict the importation of OSPF ASE routes when a router is functioning as an AS border router. This is accomplished by specifying an `export ospfase` clause. Specification of an empty `export` clause can be used to restrict importation of ASEs when no ASEs are being exported. (For more information about exporting ASEs, see "Exporting to OSPF ASE and NSSA" on page 151 of *Configuring GateD*.)

Like the other interior gateway protocols, `preference` cannot be used to choose between OSPF ASE routes. That is done by the OSPF costs. Routes that are rejected by policy are stored in the table with a negative preference.

Defaults

```
import proto ospfase {  
    all ;  
};
```

Context

control statements

Examples

The following example causes no ASE routes learned via the OSPF protocol to be considered during the route selection process.

```
import proto ospfase restrict;
```

The following example causes only ASE routes learned via the OSPF protocol with tag 123 to be imported.

```
import proto ospfase tag 123 {  
    all;  
};
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

`ospf` on page 124

`preference` on page 618

import proto redirect

Name

import proto redirect- specifies ICMP redirect import policy

Syntax

```
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 ;
```

Parameters

route_filter - a set of route filters specifying particular routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

interface *interface_list* - specifies a list of interfaces to which this import policy applies. This import statement will be applied only to routes learned over the listed interfaces.

gateway *gateway_list* - specifies a list of gateways to which this import policy applies. This import statement will be applied only to routes learned via the listed gateways.

preference *preference* - specifies the preference value to be assigned to routes that match this import statement or *route_filter*

restrict - specifies that routes matching this import statement or *route_filter* are not to be considered during the route selection process

toribs *riblist* - specifies RIBs to import into

Description

Import statements specify import policy. That is, they specify the policy which is used to determine whether and how routes are to be added to the GateD internal routing table.

The importation of Redirect routes can be controlled by any source interface or source gateway.

Context

control statements

Examples

The following example installs redirects for more specific routes of 10/8 only.

```
import proto redirect {  
    10.0.0.0/8 refines ;  
};
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

preference on page 618

redirect on page 319

rip on page 69

import proto rip

Name

`import proto rip` - specifies RIP import policy

Syntax

```
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 ;
```

Parameters

route_filter - a set of route filters specifying particular routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

interface *interface_list* - specifies a list of interfaces to which this import policy applies. This import statement will be applied only to routes learned over the listed interfaces.

gateway *gateway_list* - specifies a list of gateways to which this import policy applies. This import statement will be applied only to routes learned via the listed gateways.

preference *preference* - specifies the preference value to be assigned to routes that match this import statement or *route_filter*

restrict - specifies that routes matching this import statement or *route_filter* are not to be considered during the route selection process

toribs *riblist* - specifies RIBs to import into

tag *tagvalue* - specifies the rip route tag value to which this import statement applies

Description

Import statements specify import policy. That is, they specify the policy which is used to determine whether and how routes are to be added to the GateD internal routing table.

The importation of RIP routes can be controlled by any rip tag value, source interface, or source gateway.

Context

control statements

Examples

Example 1

The following example imports, from RIP, all routes to the 10/8 network or any of its sub-nets. It restricts all routes to the 192.168/16 network, and it imports, with a preference of 110, all other routes learned via RIP:

```
import proto rip {  
    10/8 ;  
    192.168/16 exact restrict ;  
    all preference 110 ;  
} ;
```

Example 2

The following example imports, from RIP, all routes learned over the fxp0 interface:

```
import proto rip interface fxp0 {  
    all;  
};
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

preference on page 618

redirect on page 319

rip on page 69

import proto ripng

Name

`import proto ripng` - specifies RIPng import policy

Syntax

```
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 ;
```

Parameters

route_filter - a set of route filters specifying particular routes to match. These filters will specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

interface *interface_list* - specifies a list of interfaces to which this import policy applies. This import statement will be applied only to routes learned over the listed interfaces.

gateway *gateway_list* - specifies a list of gateways to which this import policy applies. This import statement will be applied only to routes learned via the listed gateways.

preference *preference* - specifies the preference value to be assigned to routes that match this import statement or *route_filter*

restrict - specifies that routes matching this import statement or *route_filter* are not to be considered in the selection process

toribs *riblist* - specifies RIBs to import into

tag *tagvalue* - specifies that this import statement applies to routes learned with the tag *tagvalue*

Description

This statement configures import policy for RIPng. The configuration syntax and options are very similar to those used by RIP.

Context

control statements

Examples

The following example imports, from RIPng, all routes learned over the fxp0 interface.

```
import proto ripng interface fxp0 {  
    all ;  
};
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*
preference on page 618

preference

Name

preference - specifies the preference assigned to a set of routes

Syntax

preference *preference*

Parameters

preference *preference* - specifies the preference value used when comparing this route to other routes from other protocols. The route to a given destination with the lowest preference at any given time becomes the active route. The active route is installed in the forwarding table and is eligible to be exported to other protocols. The default preferences are configured by the individual protocols.

Description

Preference statements are used to assign a preference value to routes. The preference value is the first tie-breaker when determining which route to a given destination will be used. Values must be in the range 0-255.

Defaults

protocol dependent

Context

import statements

route_filter

Examples

The following example causes routes to the 10/8 network, or any of its subnets, to be installed with a preference of 100, and all other routes learned from RIP to be installed with the RIP default preference:

```
import proto rip {
    10/8 preference 100;
    all;
};
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

import proto bgp on page 607

import proto ospf on page 610

import proto rip on page 614

restrict

Name

`restrict` - specifies that the routes are not desired in the routing table

Syntax

`restrict`

Parameters

none

Description

`restrict` specifies that the routes are not desired in the routing table. In some cases, the routes are not installed in the routing table. In other cases, the routes are installed with a negative preference. This prevents them from becoming active so they will not be installed in the forwarding table or exported to other protocols.

`restrict` can be specified on an `import proto` statement, in which case all routes matched by this import statement will be restricted. It can also be specified in `route_filter`, in which case only routes matching the `route_filter` will be restricted.

Defaults

none

Context

`import` statement

`route_filter`

Examples

The following example causes route learned from RIP to not be considered for route selection:

```
import proto rip restrict;
```

See Also

"Chapter 31 Route Importation" on page 137 in *Configuring GateD*

`import proto bgp` on page 607

`import proto ospfase` on page 610

`import proto rip` on page 614

tag

Name

`tag` - specifies the protocol-specific tag value to which an `import proto` statement applies

Syntax

```
tag tagvalue
```

Parameters

`tagvalue` - an integer from 0 to 65535 for RIP and RIPng routes and 0 to 4294967295 for OSPF routes.

Description

Some protocols (RIP, RIPng, and OSPF) allow the setting of a tag value that can be associated with an advertised route. This tag can then be used as an import filter for incoming routes. `tag` specifies that this import statement applies to routes learned with a matching tag. RIP and RIPng allow 16-bit tags, and OSPF allows 32-bit tags to be advertised with routes. Policy configured with `tag` applies only to routes matching the given tag.

Defaults

none

Context

```
import proto rip statement
import proto ripng statement
import proto ospfase statement
```

Examples

```
import proto rip tag 100 {
    all ;
} ;
```

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

`import proto ospfase` on page 610

`import proto rip` on page 614

`import proto ripng` on page 616

toribs

Name

toribs - specifies the ribs into which a route should be imported

Syntax

```
import proto ( ... import_parameters ... ) toribs riblist
    route_filter toribs riblist ;
```

Parameters

import_parameters - any other import parameters as specified in a given **import** command.

route_filter - a set of route filters specifying particular routes to match. These filters specify whether a route is to be accepted, and if so, with what attributes. See "Chapter 28 Route Filtering" on page 129 of *Configuring GateD* for more information.

riblist - one of the following:

unicast - specifies that the routes are to be imported into the unicast RIB

multicast - specifies that the routes are to be imported into the multicast RIB

unicast multicast - specifies that the routes are to be imported into both the unicast and multicast RIBs

Description

These parameters are used to specify into which ribs a route should be imported. This is useful when you are running PIM-SM, which does not maintain its own routing table. Instead, it relies on a unicast routing protocol to generate the multicast routing table, which it will use for performing reverse path forwarding.

toribs specified on a matching *route_filter* overrides any **toribs** specified on the **import proto** statement.

Defaults

The default varies based on protocol. MPBGP learns rib information along with routes, and places each route into the appropriate ribs. All other import protocols place routes into the unicast routing table only, by default.

Context

import statement

route_filter

Examples

Example 1

The following example imports all routes learned from RIP in to both the unicast rib and the multicast rib.

```
import proto rip toribs unicast multicast {  
    all;  
};
```

Example 2

The following example does the same thing, but uses the *route_filter* version.

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

See Also

“Chapter 31 Route Importation” on page 137 in *Configuring GateD*

`import proto bgp` on page 607

`import proto ospfase` on page 610

`import proto rip` on page 614