Chapter 29
Route Aggregation and Generation

aggregate

Name

aggregate - creates a summary (aggregate) route from more specific contributor routes

Syntax

```
aggregate ( [ inet6 ] default |
  network ( mask mask | masklen masklen | / masklen ) )
    [ preference preference_value ]
    [ bgp ]
    [ brief ]
    [ [ toribs ] ( unicast | multicast | unicast multicast ) ]
    {
        aggregate_list
    };
```

Parameters

[inet6] default - sets the prefix for the aggregate to 0.0.0.0/0 or ::/0 if **inet6** is specified

network - the address of the aggregate, in dotted-quad format (xxx.xxx.xxx.xxx) or IPv6 format

mask - the address mask in dotted-quad format (xxx.xxx.xxx.xxx) or IPv6 format

masklen - the number of contiguous one bits at the beginning of the mask

preference_value - a preference number (integer) from 0 to 255, inclusive

bgp - specifies that this aggregate will use BGP rules to determine whether or not to include each route

brief - specifies that the AS path should be truncated to the longest common AS path

toribs - specifies that the aggregate is restricted to this RIB

aggregate_list - one or more of the following: **proto bgp**, **proto rip**, **proto ripng**, **proto ospf**, **proto ospfase**, **proto direct**, **proto static**, **proto kernel**, **proto isis**, **proto aggregate**, or **proto all** as defined later in this document

Description

Route aggregation is a method of generating a more general route, given the presence of a specific route. It is used, for example, at an autonomous system border to generate a route to a network to be advertised via BGP, given the presence of one or more subnets of that network learned via RIP. No aggregation is performed unless explicitly requested in an aggregate statement.

Defaults

By default, an aggregate applies to all RIBs to which any contributing route applies. For example, an aggregate applies to the unicast RIB if and only if any contributing route applies to the unicast RIB.

Context

`global` statement

Examples

Example 1

```
aggregate 10/8 {
    proto bgp aspath “(64512 .*)” origin any {
        10/8 refines;
    };
};
```

Example 2

```
aggregate 10.0.0.0 masklen 8 {
    proto static {
        10.0.0.0 masklen 8 refines;
    };
};
```

See Also

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

as

Name

as - restricts selection of routes to those learned from the specified autonomous system

Syntax

as *ASN*

Parameters

ASN - the autonomous system number from which routes are to be learned

Description

as restricts selection of routes to those learned from the specified autonomous system.

Defaults

none

Context

aggregate proto bgp statement

generate proto bgp statement

Examples

```
aggregate 10/8 bgp {
    proto bgp as 64512 {
        10/8 refines;
    };
};
```

See Also

aggregate on page 673

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

generate on page 678

proto bgp on page 686

aspath

Name

aspath - restricts selection of routes to those that match the specified AS path

Syntax

aspath *aspath-regular-expression*

Parameters

aspath-regular-expression - an AS path regular expression. The syntax of these regular expressions is described in “AS Path Regular Expressions” on page 131, and in Section 4.2 of RFC 1164.

Description

aspath restricts selection of routes to those that match the specified AS path.

Defaults

none

Context

aggregate proto bgp statement

generate proto bgp statement

Examples

```
aggregate 10/8  {
    proto bgp aspath “(64512 .*)” origin any {
        10/8 refines;
    };
};
```

See Also

aggregate on page 673

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

generate on page 678

proto bgp on page 686

brief

Name

`brief` - specifies that the AS path should be truncated to the longest common AS path

Syntax

`brief`

Parameters

none

Description

`brief` specifies that the AS path should be truncated to the longest common AS path.

Defaults

The default is to build an AS path consisting of SETs and SEQUENCEs of all contributing AS paths.

Context

`aggregate` statement

Examples

```
aggregate 10/8 bgp brief {
    proto bgp {
        10/8 refines;
    };
};
```

See Also

`aggregate` on page 673

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

generate

Name

`generate` - creates a summary (aggregate) route from one or more specific component routes

Syntax

```
generate ( [ inet6 ] default |
  network ( mask mask | masklen masklen | / masklen ) )
    [ preference preference_value ]
    [ [ toribs ] ( unicast | multicast | unicast multicast ) ]
    [ noinstall ]
    {
        aggregate_list
    };
```

Parameters

`[ inet6 ] default` - sets the prefix for the generate to 0.0.0.0/0 or ::/0 if **`inet6`** is specified

`network` - the generate address, in dotted-quad format (xxx.xxx.xxx.xxx) or IPv6 format

`mask` - the address mask (modification) in dotted-quad format (xxx.xxx.xxx.xxx) or IPv6 format

`masklen` - the number of contiguous one bits at the beginning of the mask

`preference_value` - a preference number (integer) from 0 to 255, inclusive

`noinstall` - specifies that the generated route is not installed in the FIB

`toribs` - specifies that the aggregate is restricted to this RIB

`aggregate_list` - one or more of the following: **`proto bgp`**, **`proto rip`**, **`proto ripng`**, **`proto ospf`**, **`proto ospfase`**, **`proto direct`**, **`proto static`**, **`proto kernel`**, **`proto isis`**, **`proto aggregate`**, or **`proto all`** as defined later in this document

Description

Route generation is a method of generating a more general route, given the presence of a specific route. **`generate`** differs from **`aggregate`** only in terms of the route installed in the kernel. **`generate`** causes a route to the specified generate address to be installed if any contributor route is active. Its primary use is to generate a default route.

Defaults

By default, generate applies to all RIBs to which any contributing route applies. For example, a generate applies to the unicast RIB if and only if any contributing route applies to the unicast RIB.

Context

`global` statement

Examples

Example 1

```
generate 10/8 {
    proto bgp aspath “(64512 .*)” origin any {
        10/8 refines;
    };
};
```

Example 2

```
generate 10.0.0.0 masklen 8 {
    proto static {
        10.0.0.0 masklen 8 refines;
    };
};
```

See Also

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

preference

Name

`preference` - specifies the aggregation preference value to be assigned to a contributor route

Syntax

`preference` *`preference_value`*

Parameters

`preference_value` - a preference number (integer) from 0 to 255, inclusive

Description

`preference` specifies the aggregation preference for a contributing aggregate route. Routes that match the route filters are called "contributing" routes. They are ordered on the list of contributing routes for a given aggregate route according to the aggregation preference value that applies to them, with the lowest (best) preference values coming first. The aggregation preference value can be specified on the **`aggregate`** or **`generate`** statement, any **`aggregate`** source statement, or any *`route_filters`*. In addition to being used to order contributing routes, the aggregation preference value associated with the first contributor on this ordered list is used as the route preference value for the aggregate route itself.

When ordering contributing routes on the contributor list, if two contributors have the same aggregation preference value, then the contributor with the lowest route preference is ordered before the other contributors with the same aggregation preference value.

In the case of **`generate`**, in addition to using the aggregation preference value of the first contributor in the list as the aggregate route's preference, the nexthops used for the aggregate are taken to be those of the first contributor in the list.

The aspath for an aggregate is formed by combining the aspath of each contributor. If BGP rules are configured for aggregation, the contributor order can impact the MED and nexthop values in the combined aspath generated for the aggregate. This is due to the fact that the values of MED and nexthop from the first valid BGP contributor route encountered while traversing the contributor list in order are used in the combined aspath.

Defaults

```
preference 130;
```

Context

`aggregate` statement

`generate` statement

`route_filter`

Examples

```
    aggregate 10/8 preference 130 bgp {
        proto bgp {
            10/8 refines;
        };
    };
```

See Also

`aggregate` on page 673

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

`generate` on page 678

proto aggregate

Name

`proto aggregate` - filters on routes that are aggregates of other routes

Syntax

```
proto aggregate restrict ;
proto aggregate
    [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol- or *`route_filter`*-specific.

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See "Chapter 28 Route Filtering" on page 129 of the *GateD Configuration Guide* for additional information.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto aggregate`** or *`route_filter`*

Description

The **`proto aggregate`** aggregate source statement is used to match routes that have been aggregated from other routes for inclusion in an aggregate route.

In the **`restrict`** version of this aggregate source statement, any aggregated routes are prohibited from inclusion in the associated aggregation target command.

If any *`route_filters`* are provided, then any routes matching those filters will be included in the aggregate for which this **`proto aggregate`** is an aggregate source. Any number of **`proto aggregate`** sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto aggregate`** export source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example aggregates only certain other aggregates into a more general aggregate.

```
aggregate 10/14 {
```

```
          proto aggregate {
              10.1/16;
              10.2/16;
      } ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto all

Name

`proto all` - filters on routes learned via any source

Syntax

```
proto all restrict ;
proto all
    [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol- or *`route_filter`*-specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto all`** or *`route_filter`*

Description

The **`proto all`** aggregate source statement is used to match routes from any source for inclusion in the associated aggregate route. The primary use of this aggregate source statement is to include all potential contributors into an aggregate statement. Protocol-specific aggregate source statements with **`restrict`** can then be used to limit the set of actual contributors to the aggregate.

In the **`restrict`** version of this aggregate source statement, no routes from any source will be included in the associated aggregate.

If any *`route_filter`*s are provided, then any routes matching those filters will be included in the aggregate for which this **`proto all`** is an aggregate source. Any number of **`proto all`** sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto all`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

This example includes all contributors, except those from kernel or static in the 10/8 aggregate.

```
aggregate 10/8 {
    proto kernel restrict ;
    proto static restrict ;
    proto all {
        10/8 refines ;
    } ;
} ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto bgp

Name

`proto bgp` - filters on routes learned via BGP

Syntax

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

Parameters

ASN - autonomous system number from 1 to 65535

aspath-regular-expression - The syntax of these regular expressions is described in "AS Path Regular Expressions" on page 131, and in Section 4.2 of RFC 1164.

preference *preference* - specifies the aggregation preference for routes that match the corresponding **proto bgp** or *route_filter*

route_filter - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See "Chapter 28 Route Filtering" on page 129 of the *GateD Configuration Guide* for additional information.

restrict - used to specify that contributor routes are to be excluded from an aggregate. **restrict** can be protocol- or *route_filter*-specific.

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

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

Description

The **proto bgp** aggregate source statement is used to match routes learned via the BGP protocol for inclusion in an aggregate route.

The statement includes an optional **as** or **aspath** component. If any *route_filter*s are provided, then any routes matching those filters which also match the **as** portion will be included in the aggregate route for which this **proto bgp** is the source. Any number of

proto bgp aggregate sources can be used within the context of a given aggregate statement, so long as they are not repeated. **restrict** specified on a *route_filter* indicates matching routes are not to be contributors.

Defaults

By default, there are no **proto bgp** aggregate source statements.

Context

aggregate statement

generate statement

Examples

```
aggregate 192.2.0.0 masklen 24 {
    proto bgp as 65412 {
        192.2.0.0/24 refines ;
    } ;
} ;
```

See Also

Application of the Border Gateway Protocol in the Internet (RFC 1164) at http://ietf.org/rfc/rfc1164.txt

“AS Path Regular Expressions” on page 131 in *Configuring GateD*

“Chapter 30 BGP Communities” on page 133 in *Configuring GateD*

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

proto direct

Name

`proto direct` - filter on directly connected interfaces

Syntax

```
proto direct restrict ;
proto direct
    [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See "Chapter 28 Route Filtering" on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol- or *`route_filter`*-specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto direct`** or *`route_filter`*

Description

The **`proto direct`** aggregate source statement is used to match routes associated with directly connected interfaces for inclusion in an aggregate route.

In the **`restrict`** version of this aggregate source statement, no direct routes will be included in the aggregate in which this **`proto direct`** occurs.

If any *`route_filter`*s are provided, then any routes matching those filters will be included in the aggregate for which this **`proto direct`** is a source. Any number of **`proto direct`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates matching routes are not to be contributors.

Defaults

By default, no **`proto direct`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example includes all direct (interface) addresses that are more specific than 10/16, into a 10/16 aggregate route.

```
aggregate 10/16 {
```

```
            proto direct {
                10/16 refines ;
            } ;
        } ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto isis

Name

`proto isis` - filters on routes learned via IS-IS

Syntax

```
proto isis restrict ;
proto isis
    [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto isis`** or *`route_filter`*

Description

The **`proto isis`** aggregate source statement is used to match routes that have been propagated through IS-IS for inclusion in an aggregate route.

In the **`restrict`** version of this aggregate source statement, no IS-IS routes will be included in the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters will be included in the aggregate for which this **`proto isis`** is an aggregate source. Any number of **`proto isis`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto isis`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example aggregates all more specific routes of 10./16 except for 10.1.1.42, if learned from IS-IS.

```
aggregate 10.1/16 {
        proto all {
            10.1/16 refines ;
        } ;
        proto isis {
            10.1.1.42 restrict ;
        } ;
    } ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto kernel

Name

`proto kernel` - filters on routes that are learned from the FIB

Syntax

```
proto kernel restrict ;
proto kernel
    [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See "Chapter 28 Route Filtering" on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto kernel`** or *`route_filter`*

Description

The **`proto kernel`** aggregate source statement is nearly identical to the **`proto static`** aggregate source statement. The only difference is that, rather than referring to routes statically configured by GateD, this statement refers to routes that are learned from the FIB. Of course, it is not normally expected that routes will be learned from the FIB; however, there are two cases where this is possible. First, the routes could be remnant routes learned via the route socket after a software crash. Second, another administrative authority may have directly added routes to the FIB. This is most commonly the case when GateD is running on a UNIX system (in which case the FIB is the kernel RIB), and the super user adds routes via the route command.

In the **`restrict`** version of this aggregate source statement, no kernel routes will be included in the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters will be included in the aggregate for which this **`proto kernel`** is an aggregate source. Any number of **`proto kernel`** aggregate sources can be used within the context of a single aggregate command.

Defaults

By default, no **`proto kernel`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example aggregates any more specific 223/8 routes learned from the kernel with the exception of 223.1/16.

```
aggregate 223/8 {
    proto kernel {
        all ;
        223.1/16 exact restrict ;
    } ;
} ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto ospf

Name

`proto ospf` - filter on routes learned via OSPF

Syntax

```
proto ospf [ tag tagvalue ] restrict ;
proto ospf [ tag tagvalue ] [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`tagvalue` - an arbitrary number from 0 to 4,294,967,295

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto ospf`** or *`route_filter`*

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See "Chapter 28 Route Filtering" on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

Description

The **`proto ospf`** aggregate source statement is used to match routes that have been propagated through OSPF for inclusion in the associated aggregate route. It deals only with routes learned directly from OSPF and does not consider routes that are AS external to OSPF. For these routes, see the **`proto ospfase`** aggregate source statement.

In the **`restrict`** version of this aggregate source statement, no OSPF routes that match the optional *`tag`* (or no OSPF routes at all in the absence of the tag parameter) will be included in the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters that also match the optional tag parameter will be included in the aggregate route for which this **`proto ospf`** is an aggregate source. Any number of **`proto ospf`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates matching routes are not to be contributors.

Defaults

By default, no **`proto ospf`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example includes all OSPF routes with tag 1234 that are more specific than 10/8 in the aggregate.

```
aggregate 10/8 {
    proto ospf tag 1234 {
        all ;
    } ;
} ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto ospfase

Name

`proto ospfase` - filters on routes learned via OSPF that are AS External

Syntax

```
proto ospfase [ tag tagvalue ] restrict ;
proto ospfase [ tag tagvalue ] [ preference preference ]
    { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`tagvalue` - an arbitrary number from 0 to 4,294,967,295

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto ospfase`** or *`route_filter`*

Description

The **`proto ospfase`** aggregate source statement is used to match routes that have been propagated through OSPF as ASE for inclusion in the associated aggregate route. To aggregate routes learned as OSPF routes, see **`proto ospf`**.

In the **`restrict`** version of this aggregate source statement, no OSPF ASE routes that match the optional *`tag`* (or no OSPF ASE routes at all in the absence of the tag parameter) will be included in the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters that also match the optional *`tag`* parameter will be included in the aggregate route for which this **`proto ospf`** is an aggregate source. Any number of **`proto ospfase`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto ospfase`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

The following example includes all OSPF ASE routes with tag 1234 that are more specific than 10/8 in the aggregate.

```
aggregate 10/8 {
    proto ospfase tag 1234 {
        all ;
    } ;
} ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto rip

Name

`proto rip` - filter on routes learned via RIP

Syntax

```
 proto rip
     [ tag tagvalue ]
     restrict ;
 proto rip
     [ tag tagvalue ]
     [ preference preference ]
     { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`tagvalue` - an arbitrary number from 0 to 65535

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto rip`** or *`route_filter`*

Description

The **`proto rip`** aggregate source statement is used to match RIP routes for inclusion in the associated aggregate route.

If the optional **`tag`** parameter is used, only routes matching the *`tagvalue`* will be considered for aggregation.

In the **`restrict`** version of this aggregate source statement, any RIP routes that match the optional **`tag`** parameter (or all RIP routes in the absence of the **`tag`**) will be excluded from the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters which also match the optional tag parameter will be included in the aggregate route for which this **`proto rip`** is an aggregate source. Any number of **`proto rip`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto rip`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

```
aggregate 10.0.0.0 masklen 8 {
    proto rip {
        10/8 ;
    } ;
} ;
```

See Also

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*

proto ripng

Name

`proto ripng` - filter on routes learned via RIPng

Syntax

```
 proto ripng
     [ tag tagvalue ]
     restrict ;
 proto ripng
     [ tag tagvalue ]
     [ preference preference ]
     { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`tagvalue` - an arbitrary number from 0 to 65535

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto ripng`** or *`route_filter`*

Description

The **`proto ripng`** aggregate source statement is used to match RIPng routes for inclusion in the associated aggregate route.

If the optional **`tag`** parameter is used, only routes matching the *`tagvalue`* will be considered for aggregation.

In the **`restrict`** version of this aggregate source statement, any RIPng routes that match the optional **`tag`** parameter (or all RIPng routes in the absence of the **`tag`**) will be excluded from the associated aggregate route.

If any *`route_filter`*s are provided, then any routes matching those filters that also match the optional tag parameter will be included in the aggregate route for which this **`proto ripng`** is an aggregate source. Any number of **`proto ripng`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates that matching routes are not to be contributors.

Defaults

By default, no **`proto ripng`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

```
aggregate feC0:: masklen 64 {
    proto ripng {
        feC0:: masklen 64 ;
    } ;
} ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

proto static

Name

`proto static` - filter on statically configured routes

Syntax

```
 proto static restrict ;
 proto static
     [ preference preference ]
     { [ route_filter [ preference preference | restrict ] ; ] } ;
```

Parameters

`route_filter` - a set of route filters used to select particular routes or sets of routes for inclusion in or exclusion from an aggregate route. See “Chapter 28 Route Filtering” on page 129 of the *GateD Configuration Guide* for additional information.

`restrict` - used to specify that contributor routes are to be excluded from an aggregate. **`restrict`** can be protocol or *`route_filter`* specific.

`preference` *`preference`* - specifies the aggregation preference for routes that match the corresponding **`proto static`** or *`route_filter`*

Description

The **`proto static`** aggregate source statement is used to match routes that have been statically configured for inclusion in the associated aggregate route.

In the **`restrict`** version of this aggregate source statement, no static routes will be included in the associated aggregate route.

If any *`route_filters`* are provided, then any routes matching those filters will be included in the aggregate route for which this **`proto static`** is an aggregate source. Any number of **`proto static`** aggregate sources can be used within the context of a single aggregate command. **`restrict`** specified on a *`route_filter`* indicates matching routes are not to be contributors.

Defaults

By default, no **`proto static`** aggregate source exists.

Context

`aggregate` statement

`generate` statement

Examples

```
      aggregate 10/8 {
          proto static {
              10/8 ;
```

```
        } ;
    } ;
```

See Also

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

restrict

Name

`restrict` - specifies routes that are not to be considered as contributors of the specified aggregate

Syntax

`restrict`

Parameters

none

Description

`restrict` specifies routes that are not to be considered as contributors of the specified aggregate. **`restrict`** can be used to restrict contributors on a protocol and/or prefix basis.

Defaults

none

Context

`aggregate proto` statement

`generate proto` statement

`route_filter`

Examples

```
aggregate 10/8 {
    proto static restrict;
    proto all {
        10/8 refines;
        10.1.2.3 restrict;
    };
};
```

See Also

`aggregate` on page 673

"Chapter 33 Route Aggregation and Generation" on page 155 in *Configuring GateD*

`generate` on page 678

toribs unicast | multicast

Name

`toribs` - specifies that the aggregate is restricted to this RIB

Syntax

```
toribs ( unicast | multicast | unicast multicast )
```

Parameters

none

Description

`toribs unicast` specifies that the aggregate is restricted to the unicast RIB. The default is all RIBs (unicast and multicast). **`toribs multicast`** specifies that the aggregate is restricted to the multicast RIB.

Defaults

The default is all RIBs (unicast and multicast).

Context

`aggregate` statement

Examples

Example 1

In this example, only “10/8 refines” routes in the multicast RIB contribute to the 10/8 multicast aggregate route.

```
aggregate 10/8 bgp toribs multicast {
    proto bgp {
        10/8 refines;
    };
};
```

Example 2

```
aggregate 10/8 bgp toribs unicast multicast {
    proto bgp {
        10/8 refines;
    };
};
```

See Also

`aggregate` on page 673

“Chapter 33 Route Aggregation and Generation” on page 155 in *Configuring GateD*