Chapter 26
BGP Communities

comm

Name

`comm` - specifies a set of communities to match

Syntax

`comm {` *`community_list`* **`}`**

Parameters

`{` *`community_list`* **`}`** - specifies the set of communities that are to be matched

Description

The **`comm`** clause is used to specify a set of communities that must be matched for a route to be considered a match.

Defaults

The default is to ignore the communities attribute.

Context

BGP import statements or BGP subclauses of export statements

Examples

The following example exports to BGP peers in AS 65534 all routes that were learned from BGP peers in AS 65533 with the community 65533 101:

```
export proto bgp as 65534 {
    proto bgp as 65533 comm { comm-split 65533 101; } {
        all;
    };
};
```

See Also

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

`comm-add` on page 595

`comm-delete` on page 596

`community_list` on page 597

`export` statements on page 623

`import` statements on page 605

comm-add

Name

`comm-add` - specifies a set of communities to be added on export

Syntax

`comm-add {` *`community_list`* **`}`**

Parameters

`{` *`community_list`* **`}`** - specifies the set of communities to be added to routes matching this export clause

Description

The **`comm-add`** clause is used to specify a set of communities that are to be added to route advertisements matching a particular BGP export statement.

Defaults

The default is to not add communities to outgoing route advertisements.

Context

BGP export statements

Examples

The following example adds an arbitrary community (0x123 0x231) to BGP routes learned from peers in AS 65533 when they are exported to BGP peers in AS 65534:

```
export proto bgp as 65534 comm-add { comm-hex 0x123 0x231; } {
    proto bgp as 65533 {
        all;
    };
};
```

See Also

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

`community_list` on page 597

`comm` on page 593

`comm-delete` on page 596

`export` statements on page 623

`import` statements on page 605

comm-delete

Name

`comm-delete` - specifies a set of communities to be removed on export

Syntax

`comm-delete {` *`community_list`* **`}`**

Parameters

`{` *`community_list`* **`}`** - specifies the set of communities to be deleted from routes matching this export clause

Description

The **`comm-delete`** clause is used to specify a set of communities that are to be removed from routes being advertised as a result of matching the export statement containing the **`comm-delete`** clause.

Defaults

The default is to not remove communities from out-going route advertisements.

Context

BGP export statements

Examples

The following example removes an arbitrary community (0x123 0x231) from all routes learned from BGP peers in AS 65534 when they are advertised to BGP peers in AS 65533:

```
export proto bgp as 65533 comm-delete { comm-hex 0x123 0x231; } {
    proto bgp as 65534 {
        all;
    };
};
```

See Also

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

`export` statements on page 623

`import` statements on page 605

community_list

Name

`community_list` - a list of communities that is used for matching or modifying AS paths

Syntax

`community none ;`

`community no-export ; ...`

`community no-advertise ; ...`

`community no-export-subconfed ; ...`

`comm-hex` *`hex-number hex-number`* **`; ...`**

`comm-split` *`autonomous_system community-id`* **`; ...`**

Parameters

`community none` - This parameter specifies the empty set of communities. It is useful only in the comm statement where matches are being attempted. When used in that context, if a route has any communities associated with it, then it does not match; otherwise, it does match.

`community no-export` - Specifies the well-known community NO_EXPORT as defined in RFC 1997. Routes tagged with this community are not to be exported outside of a confederation boundary.

`community no-advertise` - Specifies the well-known community NO_ADVERTISE as defined in RFC 1997. Routes tagged with this community are not to be advertised to any other peers.

`community no-export-subconfed` - Specifies the well-known community NO_EXPORT_SUBCONFED as defined in RFC 1997. Routes tagged with this community are not to be advertised to external peers, even if they are within the same confederation.

`comm-hex` *`hex-number hex-number`* - This parameter specifies any arbitrary community that is the concatenation of the two sixteen-bit numbers specified.

`comm-split` *`autonomous_system community-id`* - This parameter specifies a community that is the concatenation of the AS number ASN and the arbitrary sixteen-bit number.

Description

A *`community_list`* specifies a series of communities that are to be matched against a route, added to a route on export, or deleted from a route on export. Any number of communities can be specified in a single community list, in any order.

Defaults

none

Context

`comm-add`

comm-delete

comm

Examples

The following example exports all static routes to BGP peers in AS 65534, adding the well-known NO_EXPORT community:

```
export proto bgp as 65534 comm-add { community no-export; } {
    proto static {
        all;
    };
};
```

See Also

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

comm-add on page 595

comm-delete on page 596

comm on page 593

export statements on page 623

import statements on page 605

ext-comm

Name

`ext-comm` - specifies a set of extended communities to match

Syntax

`ext-comm {` *`extended_community_list`* **`}`**

Parameters

`extended_community_list` - specifies the set of extended communities to be matched

Description

The **`ext-comm`** clause is used to specify a set of extended communities that must be matched for a route to be considered a match.

Defaults

The default is to ignore the communities attribute.

Context

BGP import statements or BGP subclauses of export statements

Examples

The following example exports to BGP peers in AS 65534 all routes that were learned from BGP peers in AS 65533 with the extended **`route-target-as`** community 65533 101:

```
export proto bgp as 65534 {
    proto bgp as 65533 ext-comm { route-target-as 65533 101: } {
        all;
    };
};
```

See Also

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

`export` statements on page 623

`ext-comm-add` on page 600

`ext-comm-delete` on page 601

`extended_community_list` on page 602

`import` statements on page 605

ext-comm-add

Name

`ext-comm-add` - specifies a set of extended communities to be added on export

Syntax

`ext-comm-add {` *`extended_community_list`* **`}`**

Parameters

`extended_community_list` - specifies the set of extended communities to be added to routes matching this export clause

Description

The **`ext-comm-add`** clause is used to specify a set of extended communities that are to be added to route advertisements matching a particular BGP export statement.

Defaults

The default is to not add extended communities to outgoing route advertisements.

Context

BGP export statements

Examples

The following example adds the **`route-target-ip`** extended community to BGP routes learned from peers in AS 65533 when they are exported to BGP peers in AS 65534:

```
export proto bgp as 65534 ext-comm-add { route-target-ip 192.0.2.1 100; } {
    proto bgp as 65533 {
        all;
    };
};
```

See Also

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

`export` statements on page 623

`extended_community_list` on page 602

`ext-comm-delete` on page 601

`ext-comm` on page 599

`import` statements on page 605

ext-comm-delete

Name

ext-comm-delete - specifies a set of extended communities to be removed on export

Syntax

ext-comm-delete { *extended_community_list* **}**

Parameters

extended_community_list - specifies the set of extended communities to be deleted from routes matching this export clause

Description

The **ext-comm-delete** clause is used to specify a set of communities that are to be removed from routes being advertised as a result of matching the export statement containing the **ext-comm-delete** clause.

Defaults

The default is to not remove extended communities from outgoing route advertisements.

Context

BGP export statements

Examples

The following example removes the **route-origin-as** extended community from all routes learned from BGP peers in AS 65534 when they are advertised to BGP peers in AS 65533.

```
export proto bgp as 65533 ext-comm-delete { route-origin-as 65533 314; } {
    proto bgp as 65534 {
        all;
    };
};
```

See Also

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

export statements on page 623

import statements on page 605

extended_community_list

Name

extended_community_list - a list of extended communities used for matching or modifying AS paths

Syntax

route-target-as *autonomous_system_number 4_byte_number* ;

route-target-ip *ip_address 2_byte_number* ;

route-origin-as *autonomous_system_number 4_byte_number* ;

route-origin-ip *ip_address 2_byte_number* ;

Parameters

autonomous_system_number - the autonomous system number of this router

2_byte_number - a positive number in standard GateD number format with a maximum value of 2^{16} - 1

4_byte_number - a positive number in standard GateD number format with a maximum value of 2^{32} - 1

Description

An *extended_community_list* specifies a series of extended communities that are to be matched against a route, added to a route on export, or deleted from a route on export. Any number of extended communities can be specified in a single extended community list in any order. No extended community can appear more than once in this list.

Defaults

none

Context

ext-comm-add

ext-comm-delete

ext-comm

Examples

The following example exports all static routes to BGP peers in AS 65534, adding the **route-origin-ip** extended community with the IP address of 192.0.2.1 and a value of 500:

```
export proto bgp as 65534 ext-comm-add { route-origin-ip 192.0.2.1 500; } {
    proto static {
        all;
    };
};
```

See Also

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

`export` statements on page 623

`ext-comm-add` on page 600

`ext-comm-delete` on page 601

`ext-comm` on page 599

`import` statements on page 605