

# GateD Configuration—GTDCFGxx

---

This chapter outlines the configuration information needed to configure the GateD routing protocol. It contains these sections:

- GateD Configuration Member—(GTDCFGxx)  
Introduces the GateD configuration member
- Statement Classes  
Describes GateD statement classes
- Statement Primitives  
Describes the GateD statement primitives
- Directive Statements  
Describes the GateD directive statements
- Trace Statements  
Describes the GateD trace statements
- Options Statements  
Describes the GateD options statements
- Interfaces Statements  
Describes the GateD interfaces statements
- Definition Statements  
Describes the GateD definition statement.
- Protocol Statements  
Describes the GateD protocol statements
- Route Statements  
Describes the Route protocol statements
- Control Statements  
Describes the Control protocol statements

## GateD Configuration Member—(GTDCFGxx)

Use the GTDCFGxx member to specify configuration parameters for the GateD routing protocol. To activate GateD, specify GATED(xxxx) on the IP statement, where xxxx is the GTDCFGxx member you wish to use.

The GateD configuration member is very different from most other configuration files for Cisco IOS for S/390. It consists of a sequence of statements, each terminated by a semi-colon (;). Statements are composed of tokens separated by white space (any combination of blanks, tabs, and newlines). Comments use the “C style” comment which starts with a “/\*” and continues until it reaches “\*/”.

---

**Note** Multicasting and SNMP are *not* supported by GateD in Cisco IOS for S/390, although these features are discussed in this section.

---

GateD relies heavily on functions and features native to UNIX operating systems. These functions and features are emulated on MVS by Cisco IOS for S/390 or by the SAS/C runtime library. References to the UNIX kernel in this section refer to Cisco IOS for S/390 itself and not the MVS operating system. For information about various function routines generally native to UNIX read:

- *Cisco IOS for S/390 C/Socket Programmer's Reference*
- *Cisco IOS for S/390 RPC/XDR Programmer's Reference*
- *SAS/C Library Reference, Second Edition, Volumes 1 and 2*
- The man pages on UNIX. Note that the man pages are UNIX system dependent and may not be the same under all UNIX implementations.

Virtual IP Addressing is available with Fault Tolerant operation. For more information, read *Virtual IP Addressing*, in Fault Tolerant Network Configuration.

## Statement Classes

There are eight classes of statements. The first two classes, directives and trace, may be specified in the configuration file in any order. The remaining six classes must be specified in the order shown.

### directives

These statements are immediately acted upon by the parser. They are used to specify included files and the directory in which they reside. Unlike other statements which terminate with a semi-colon (;), directive statements terminate with a newline.

### trace

These statements control tracing options.

### options

These statements allow specification of some global options.

### interface

These statements specify interface options.

---

### definition

These statements specify options, the autonomous system and martian networks.

### protocol

These statements enable or disable protocols and set protocol options.

### route

Static routes are defined by route statements.

### control

Control statements define routes that are imported from routing peers and routes that are exported to these peers.

## Statement Primitives

The primitives that can be used in the GateD statements are described below and outlined in the sections that follow.

### host

Any host. A host may be specified by its IP address or by a domain name. If a domain name is specified that has multiple IP addresses, it is considered an error. The host bits in the IP address must be non-zero.

### network

Any network. A network may be specified by its IP address or a network name. The host bits in a network specification must be zero. default may also be used to specify the default network (0.0.0.0).

### destination

Any host or network.

### dest\_mask

Any host or network with an optional mask:

```
all
network
network mask mask
network mask-length bits
host host
```

A mask is a dotted quad specifying which bits of the destination are significant. All may be used to specify that any IP address may be matched. The number of contiguous bits may be used instead of an explicit mask.

**autonomous system**

A number between 1 and 65534 assigned by the Internet Assigned Numbers Authority to represent an autonomous system.

**gateway**

A gateway must be a host on an attached network.

**interface**

An interface can be specified by IP address, domain name, or interface name.

Interface names for Cisco IOS for S/390 are assigned as follows:

**Table 17-1 Interface Names for Cisco IOS for S/390**

fd0	FDDI
tr0	Token Ring
hy0	HYPERchannel
en0	Ethernet.
cl0	CLAW
vr0	Virtual

Additional interfaces have the same name incremented by 1 (in other words, a second Ethernet interface would be en1).

To force GateD to use a specific device, you must use the routerid statement. Read Definition Statements for more information.

**gateway\_list**

A gateway list is a list of one or more gateways.

**interface\_list**

An interface list is a list of one or more interface names, wildcard names (names without a number) or addresses, or the token all, which refers to all interfaces.

**preference**

A preference is used to determine the order of routes to the same destination in the routing table. GateD allows one route to a destination per protocol/per autonomous system. In the case of multiple routes, the route to use is chosen by preference, which is a number between 0 and 255, with 0 being the most preferred and 255 being the least preferred.

In case of a preference tie, if the two routes are from the same protocol and from the same autonomous system, GateD will choose the route with the lowest metric. Otherwise GateD will choose the route with the lowest numeric next-hop gateway address.

---

**metric**

A valid metric for the specified protocol.

## Directive Statements

Use directive statements to specify included files and the directory in which they reside. Unlike other statements which terminate with a semicolon (;), directive statements terminate with a newline. Directive statements are acted upon immediately by the parser.

This statement causes the specified file to be parsed completely before resuming with this file:`%include "ddn:ddname(MEMBER)"`.

Nesting is supported up to 10 levels.

```
options gendefault;  
%include "ddn:sysparm(GTDRIP)"  
traceoptions none;
```

If a member named GTDRIP in a data set in the sysparm DD concatenation exists with this data

```
rip yes;
```

the above example will parse as follows:

```
options gendefault;  
rip yes;  
traceoptions none;
```

## Trace Statements

Trace statements control tracing options for GateD. This is how trace statements are used:

```
traceoptions traceoption [traceoption [...]]  
[except traceoption [traceoption [... ]]];
```

If *none* is the only option specified, tracing is turned off. If *except* is specified, flags listed before it are turned on and flags listed after it are turned off. Use this to turn on all but a few flags. The trace flags are described below:

<i>all</i>	Turns on all of the tracing options below except nostamp.
<i>none</i>	Turns off tracing.
<i>except</i>	A toggle mechanism; flags listed before are turned on and flags listed after are turned off.
<i>general</i>	Turns on internal, external and route.
<i>internal</i>	Turns on internal errors and informational messages.
<i>external</i>	Turns on external errors.
<i>nostamp</i>	Do not timestamp all messages in the trace file.
<i>mark</i>	Outputs a message to the trace log every 10 minutes to insure GateD is still running.
<i>task</i>	Turns on task scheduling, signal handling and packet reception.
<i>timer</i>	Turns on timer scheduling.
<i>parse</i>	Tokens the parser recognizes in the configuration file.
<i>route</i>	Changes to the GateD routing table.
<i>kernel</i>	Changes to the kernel's routing table.
<i>bgp</i>	Sends and receives Border Gateway Protocol (BGP) packets. Can be modified by update and protocol.
<i>egp</i>	Sends and receives Exterior Gateway Protocol (EGP) packets. Can be modified by update and protocol.
<i>rip</i>	Sends and receives Routing Information Protocol (RIP) packets. Can be modified by update.
<i>hello</i>	Sends and receives HELLO packets. Can be modified by update
<i>icmp</i>	Sends and receives ICMP redirect packets. Can be modified by update. Redirects processed are traced under the route option.
<i>protocol</i>	Provide messages about protocol state machine transitions when used with EGP, BGP, or OSPF.
<i>update</i>	Traces the contents of protocol packets.

---

## Options Statements

The options statement allows specification of some global GateD options. This is how options statements are used:

```
options option_list;
```

### Syntax Description

*noinstall*

Do not change kernels routing table. Useful for verifying configuration files. `gendefault` [`preference preference`] [`gateway gateway`] BGP and EGP neighbors should cause the internal generation of a default route when up. By default, this route will not be installed in the kernel's routing table, but may be announced by other protocols. If a gateway is specified, the route is installed in the kernel pointing to that gateway. Announcement is controlled by referencing the special protocol default.

*nosend*

Do not send any packets. This allows running GateD on a live network to test protocol interactions without actually participating in the routing protocols. The packet traces in the GateD log can be examined to verify that GateD is functioning properly. This is useful for RIP and HELLO. Currently this does not yet apply to BGP packets.

*noresolv*

Do not try to resolve symbolic names into IP addresses by using the host network tables or Domain Name System. This is intended for systems where a lack of routing information could cause a DNSlookup to hang.

*syslog* [*upto log\_level*] *log\_level*

Controls the amount of data GateD logs via syslog on systems where `setlogmask()` is supported. The `log_levels` and other terminology are as defined in the `setlogmask(3)` man page. The default is equivalent to `syslog upto info`. See `syslog.h` for level names in `syslog` member in `H` library.

## Interfaces Statements

Use this syntax to specify interface options:

```
interfaces {  
    options [strictifs] [scaninterval time];  
    interface interface_list interface_options ;  
    define address [broadcast broadaddr | pointopoint lcladdr]  
    [netmask netmask] [multicast];  
};
```

### Syntax Description

- options** Sets some global options related to interfaces. Options are:
- strictifs*—Indicates that it is a fatal error to reference an interface in the configuration file that is not listed in a define statement or not present when GateD is started. Without this option, a warning message will be issued and GateD will continue.
  - scaninterval time*—Specifies how often GateD scans the kernel interface list for changes. The default is every 15 seconds on most systems, 60 seconds on systems that pass interface status changes through the routing socket (in other words, BSD 4.4).  
Note GateD will also scan the interface list on receipt of a PPOST USER.
- interface** Sets interface options on the specified interfaces. An *interface\_list* is all or a list of interface names (read the warning about interface names in the interface entry of the table in Statement Primitives), domain names, or numeric addresses. *interface\_options* are:
- preference pref*—Sets the preference for routes to this interface when it is up, defaults to 0.
  - down preference pref*—Sets the preference for routes to this interface when GateD believes it to be down due to lack of received routing information, defaults to 120 seconds.
  - passive*—Prevents GateD from changing the preference of the route to this interface if it is believed to be down due to lack of received routing information.
  - simplex*—Defines an interface as unable to hear its own broadcast packets. Currently defining an interface as simplex is functionally equivalent to defining it as passive.
  - reject*—Specifies that the address loopback interfaces which match these criteria will be used as the local address when installing reject routes in the kernel. Should only be used with systems based on BSD 4.3 Tahoe or earlier which have installed a reject/blackhole pseudo interface.
  - blackhole*—Specifies that the address loopback interfaces which match these criteria will be used as the local address when installing blackhole routes in the kernel. Should only be used with systems based on BSD 4.3 Tahoe or earlier which have installed a reject/blackhhole pseudo interface.



---

**define** Defines interfaces that may not be present when GateD is started. GateD considers it an error to reference a non-existent interface in the configuration file. This clause allows specification of that interface so it can be referenced in the configuration file. Definition keywords are:

*broadcast broadaddr*—Defines the interface as broadcast capable (in other words, Ethernet and Token Ring) and specifies the broadcast address.

*pointpoint lcladdr*—Defines the interface as a point-to-point interface (in other words, SLIP and PPP) and specifies the address on the local side. For this type of interface the *interface\_addr* specifies the address of the remote host.

An interface not defined as broadcast or point-to-point is assumed to be nonbroadcast multiaccess (NBMA).

**netmask netmask** Specifies the non-standard subnet mask to be used on this interface. This is currently ignored on pointpoint interfaces.

*multicast* Specifies that the interface is multicast capable (NBMA), such as an X.25 network.

## Definition Statements

Definition statements specify options, the autonomous system, and martian networks.

This definition statement sets the autonomous system of this router to be autonomous system. This option is required if BGP or EGP are in use.

**autonomousssystem** *autonomous system;*

This definition statement sets the router identifier for use by the BGP and OSPF (Open Shortest Path First) protocols. The default is the address of the first interface encountered by GateD. The address of a non-POINTOPOINT interface is preferred over the local address of a POINTOPOINT interface and an address on a loopback interface that is not the loopback address (127.0.0.1) is most preferred. This statement forces GateD to use a specific device. (Read about the interface parameter in Statement Primitives).

**routerid** *host;*

---

**Note** It is strongly recommended that you use a routerid definition.

---

The martians statement defines a list of martian addresses about which all routing information is ignored. The *martian\_list* is a semicolon-separated list of symbolic or numeric hosts with optional masks. Read the *dest\_mask* entry of the table in Statement Primitives. The *allow* parameter can be specified to explicitly allow a subset of a range that was disallowed.

```
martians {
    martian_list
};
```

## Protocol Statements

A protocol statement enables or disables the use of a protocol and controls protocol options. These statements can be specified in any order.

For all protocols, the preference option controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems.

The default metric used when propagating routes learned from other protocols is specified with defaultmetric, which itself defaults to the highest valid metric for this protocol. For many protocols this signifies a lack of reachability.

For distance vector IGPs with no explicit connections or authentication (RIP and HELLO) and redirects (ICMP), the trustedgateways clause supplies a list of gateways providing valid routing information; routing packets from other gateways are ignored. This defaults to all gateways on the attached networks.

Routing packets can be sent not only to the remote end of point-to-point links and the broadcast address of broadcast-capable interfaces, but also to specific gateways if they are listed in a sourcegateways clause and yes or on is specified. If nobroadcast is specified, routing updates will be sent only to gateways listed in the sourcegateways clause, and not to the broadcast address.

Use the interface clause to disable transmission and reception of routing packets for a particular protocol. Override an interface clause that disables sending or receiving protocol packets for specific peers with the trustedgateways and sourcegateways clauses.

For exterior protocols (BGP, EGP), the autonomous system advertised to the peer is specified by the global autonomoussystem clause unless overridden by the asout parameter. The incoming autonomous system number is not verified unless peeras is specified.

Specifying metricout fixes the outgoing metric for all routes propagated to this peer. If the peer does not share a network, use interfaces to specify which interface address to use when communicating with this peer, and use gateway to specify the next hop to use for all routes learned from this peer. An internal default is generated when routing information is learned from a peer unless the nogendefault parameter is specified.

Any protocol can have a traceoptions clause, which enables tracing for a particular protocol, group, or peer. Table 17-2 shows the allowable protocol-specific options.

**Table 17-2 Protocol Trace Options**

all	general	task	kernel
internal	external	protocol	route
update	timer		

---

## RIP Protocol Configuration

The section describes how to configure GateD for RIP (Routing Information Protocol). If the RIP clause is not specified, the default is rip on.

```
rip yes | no | on | off [ {  
    broadcast;  
    nobroadcast;  
    nocheckzero;  
    preference preference;  
    defaultmetric metric;  
    interface interface_list [noripin] [noripout]  
        [metricin metric] [metricout metric]  
        [version 1][version 2 broadcast]  
        [authentication [none | password]];  
    ...  
    trustedgateways gateway_list;  
    sourcegateways gateway_list;  
    traceoptions traceoptions;  
} ];
```

### Syntax Description

**yes | no | on | off**

Turns RIP on or off. If yes or on is specified, RIP will assume nobroadcast if there is only one interface, and broadcast if there is more than one.

**broadcast nobroadcast**

The nobroadcast option specifies that RIP packets will only be sent to gateways listed in the sourcegateways clause, if there are any. The broadcast option specifies that RIP packets will always be generated. Using broadcast with only one interface is useful only when propagating static routes or routes learned from another protocol. This will cause data packets to travel across the same network twice, which may be tolerable in certain configurations.

**nocheckzero**

Specifies that RIP should not make sure that the reserved fields in RIP packets are zero.

**preference m** *preference*

Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default preference is 100.

<b>defaultmetric</b> <i>metric</i>	Specifies the default metric used when propagating routes learned from other protocols. Defaults to the highest valid metric for this protocol. Default metric is 16.
<b>interface</b> <i>interface_list</i> [ <b>noripin</b> ] [ <b>noripout</b> ] [ <b>metricin</b> ] [ <b>metricoutmetric</b> ] [ <b>version 1</b> ] [ <b>version 2 broadcast</b> ] [ <b>authentication</b> ] [ <b>none</b>   <b>password</b> ]	Disables transmission and reception of routing packets for a particular protocol. Overrides an interface clause that disables sending or receiving protocol packets for specific peers with <b>trustedgateways</b> and <b>sourcegateways</b> clauses. Default <b>metricout</b> is 0. Default <b>metricin</b> is the kernel interface metric plus 1 (the default RIP hop count). Default authentication type is NONE (this only applies to RIP-2 packets, and is ignored for RIP-1 packets). If a password is specified, the authentication type used will be SIMPLE. The password should be a quoted string with between zero and 16 characters.
<b>trustedgateways</b> <i>gateway_list</i>	Overrides an interface clause that disables sending or receiving protocol packets for specific peers.
<b>sourcegateways</b> <i>gateway_list</i>	Overrides an interface clause that disables sending or receiving protocol packets for specific peers.
<b>traceoptions</b>	Enables tracing for a particular protocol, group, or peer. If the version is specified as or defaults to 1, RIP version 2 packets will never be sent except in response to a v2 POLL packet. If the version is specified as 2, RIP version 2 packets will be sent to the broadcast address, unless the method is explicitly specified.

---

## OSPF Protocol Configuration

The section describes how to configure GateD for the OSPF protocol. If the OSPF clause is not specified, the default is ospf off.

The OSPF statement is shown below:

```
ospf yes | no | on | off [ {  
    [defaults {  
        preference preference ;  
        cost cost ;  
        tag [tag | as [as_tag]] ;  
        type 1|2 ;  
    }];  
    [exportlimit routes ; ]  
    [exportinterval time ; ]  
    [traceoptions traceoptions ; ]  
    [monitorauthkey authkey ; ]  
    [area area | backbone  
    {  
        authtype 0 | 1 | none | simple ;  
        stub [cost cost];  
        networks {network [mask mask] ;};  
        stubhosts {host cost cost ;};  
        interface interface [cost cost] {  
            [enable | disable] ;  
            retransmitinterval time ;  
            transitdelay time ;  
            priority priority ;  
            hellointerval time ;  
            routerdeadinterval time ;  
            authkey auth_key ;  
        } ;  
        interface interface nonbroadcast [cost cost] {  
            pollinterval time ;  
            routers {gateway [eligible] ... } ;  
            [enable | disable] ;  
            retransmitinterval time ;  
            transitdelay time ;
```

```

        priority priority ;
        hellointerval time ;
        routerdeadinterval time ;
        authkey auth_key ;
    };
;]
    virtuallink neighborid routerid transitarea area {
        [enable | disable] ;
        retransmitinterval time ;
        transitdelay time ;
        priority priority ;
        hellointerval time ;
        routerdeadinterval time ;
        authkey auth_key ;
    };
    ...
};]

```

### Syntax Description

<b>area</b> <i>area</i>   <b>backbone</b>	An area is a group of contiguous networks and attached hosts. It is a dotted quad or a number between 1 and 4294967295. Area 0 is always referred to as the backbone. An OSPF backbone is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers.
<b>auth</b> type { <b>0</b>   <b>1</b>   <b>none</b>   <b>simple</b> }	1 or simple turns on simple password authorization; 0 or none is no authorization.
<b>stub</b> [ <b>cost</b> <i>cost</i> ]	This option denotes a stub area; it is not valid with the backbone parameter.
<b>networks</b> { <b>network</b> [ <b>mask</b> <i>mask</i> ]}	Specifies the scope of the area. This can be a network or subnetwork and mask.
<b>stub</b> hosts { <b>host</b> <b>cost</b> <i>cost</i> }	Directly attached hosts and their costs.
<b>interface</b> <i>interface</i> [ <b>cost</b> <i>cost</i> ]	
[ <b>enable</b>   <b>disable</b> ]	
<b>retransmit</b> interval <i>time</i>	
<b>transit</b> delay <i>time</i>	

---

**priority** *priority*

**hellointerval** *time*

**routerdeadinterval** *time*

**pollinterval** *time*

**routers** {gateway [eligible]}

[authkey *auth\_key*]

An interface is specified with an address, a name, a wildcard name (name without any number), or all. Multiple interface clauses can be specified with different parameters; the parameters used are accumulated from the interface clauses. If a parameter is specified more than once, the instance with the most specific interface reference is used. The order of precedence is address, name, wildcard\_name, all.

**virtuallink neighborid** *routerid*

**transitarea** *area*

[enable | disable]

**retransmitinterval** *time*

**transitdelay** *time*

**priority** *priority*

**hellointerval** *time*

**routerdeadinterval** *time*

**virtuallink**

Virtual links are used by OSPF to maintain a contiguous backbone. This option is used only with the backbone parameter.

**defaults**

**preference** *preference*

Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems.

**cost** *cost*

The cost used when exporting GateD routes to OSPF as ASEs.

**tag** [tag | as [as\_tag]]

The OSPF tag (an unsigned 31-bit number) to be placed on all routes exported by GateD into OSPF. When “tag as” [as\_tag] is used, tag fields are automatically generated and the as\_tag field is assigned, if specified.

**type** {1 | 2}

GateD exports routes to OSPF as type 1 ASEs by default. Use type 2 to override the default.

**exportlimit** *routes*

Specifies how many ASEs GateD exports to OSPF at one time.

**exportinterval** *time*

Specifies (in seconds) how often GateD exports ASEs to OSPF.

<b>monitorauthkey</b> <i>authkey</i>	Used for authorization of ospfmon, the OSPF monitor command. authkey is one to eight decimal digits separated by periods, a one to eight byte hexadecimal string preceded by 0x, or a one to eight character string in double quotes.
<b>traceoptions</b>	Enable tracing for a particular protocol, group, or peer.
<b>yes   no   on   off</b>	Turns the OSPF protocol handling on or off.
The sub-parameters for the OSPF statement are:	
<b>authkey</b> <i>auth_key</i>	Used for route authorization. auth_key is ne to eight decimal digits separated by periods, a one to eight byte hexadecimal string preceded by 0x, or a one to eight character string in double quotes.
<b>cost</b> <i>cost</i>	A number between 0 and 65535 specifying an OSPF internal cost. This is specified by the network administrator based on any relevant data. Default: 1
<b>[enable   disable]</b>	Enables or disables this interface for OSPF. Default: enable
<b>hellointerval</b> <i>time</i>	Time between hellos.
<b>neighborid</b> <i>routerid</i>	For a virtual link, the host name or address of the router on the other end of the virtual link.
<b>nonbroadcast</b>	Denotes that the interface is a nonbroadcast multi-access (NBMA) device.
<b>pollinterval</b> <i>time</i>	For NBMA devices, the number of seconds between polling a dead interface.
<b>priority</b> <i>priority</i>	A number between 0 and 255 specifying the priority of becoming the designated router on this interface.
<b>retransmitinterval</b> <i>time</i>	Time between LSA retransmits.
<b>routerdeadinterval</b> <i>time</i>	Time without hellos before router is called “dead.”
<b>routers</b> {gateway [eligible]}	Lists the routers on this interface.eligible specifies that the router is eligible to be the Designated Router for the area.Used only for NBMA devices.
<b>transitarea</b> <i>area</i>	For a virtual link, the area through which the link passes.
<b>transitdelay</b> <i>time</i>	Estimate of seconds to transit a linkstate update.

### Usage Notes for OSPF

- OSPF inter and intra area are always imported into the GateD routing table with a preference of 10. It is not possible to override this. OSPF Autonomous System External (ASE) routes are imported with a preference of 150. This default can be changed with the preference keyword in the defaults section. ASE routes are imported at a rate of 100 ASEs every 1 second. These parameters can be tuned with the exportlimit and exportinterval parameters.



- 
- GateD routes are exported to OSPF as ASEs with a default cost of 0 and a type of 1. By default, the tag is calculated from the AS path of the route being exported (tag as). These can all be changed in the defaults section.
  - OSPF areas can be specified in any order, but the backbone area must be specified last.
  - Reconfiguration (SIGHUP) is currently disabled when OSPF is enabled.

## HELLO Protocol Configuration

The section describes how to configure GateD for the HELLO protocol. If the HELLO clause is not specified, the default is hello off.

---

**Note** Cisco IOS for S/390 GateD does not support HELLO.

---

```
hello yes | no | on | off [ {  
    broadcast;  
    nobroadcast;  
    preference preference;  
    defaultmetric metric;  
    interface interface_list [nohelloin] [nohelloout]  
        [metricin metric] [metricout metric] ;  
    ...  
    trustedgateways gateway_list;  
    sourcegateways gateway_list;  
    traceoptions traceoptions;  
} ] ;
```

Syntax Description

**yes | no | on | off**

Turns on the HELLO protocol for GateD.

**broadcast nobroadcast**

The nobroadcast option specifies that HELLO packets will only be sent to gateways listed in the sourcegateways clause, if there are any. The broadcast option specifies that HELLO packets will be generated. Using broadcast with only one interface is useful only when propagating static routes or routes learned from another protocol. This will cause data packets to travel across the same network twice, which may be tolerable in certain configurations.

**preference** *preference*

Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default is 90.

**defaultmetric** *metric*

Specifies the default metric used when propagating routes learned from other protocols. Defaults to the highest valid metric for this protocol. Default is 30000.

**interface** *interface\_list* [nohelloin]  
[nohelloout] [metricin] [metricout]

Default metricout is zero Default metricin is a translation of the kernel interface metric into a hello metric plus 100 (the default HELLO hop count).

**trustedgateways**

Overrides an interface clause that disables sending or receiving protocol packets for specific peers.

**sourcegateways**

Overrides an interface clause that disables sending or receiving protocol packets for specific peers.

**traceoptions**

Enables tracing for a particular protocol, group, or peer.

## EGP Protocol Configuration

The section describes how to configure GateD for EGP (External Gateway Protocol). If the EGP clause is not specified, the default is `egp off`. The `egp` statement is shown below:

---

**Note** Cisco IOS for S/390 GateD does not support EGP.

---

```

egp yes | no | on | off [ {
    [preference preference ; ]
    [defaultmetric metric ; ]
    [packetsize maxpacketsize ; ]
    [traceoptions traceoptions ; ]
    [group [peeras autonomous system]
        [localas autonomous system]

```

---

```

        [maxup number]
        [preference preference]
    {
        neighbor host
            [metricout metric]
            [nogendefault]
            [importdefault]
            [exportdefault]
            [gateway gateway]
            [lcladdr local_address]
            [sourcenet network]
            [minhello min_hello]
            [minpoll min_poll]
            [traceoptions traceoptions]
        ;
        ...
    } ;
    ...]
} ;

```

### Syntax Description

<b>yes   no   on   off</b>	Turns the EGP protocol handling on or off. Default: off
<b>preference</b>	Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default is 200.
<b>defaultmetric</b>	Specifies the default metric used when propagating routes learned from other protocols. Defaults to the highest valid metric for this protocol. Default is 255.
<b>packetsize</b>	Specifies the size, in bytes, of the largest EGP packet that will be accepted or sent.
<b>traceoptions</b>	Enables tracing for a particular protocol, group, or peer.

**group** [peeras autonomous\_system]  
[localas autonomous\_system]  
[maxup *number*]  
[preference *preference*]

Lists a group of EGP peers in one autonomous system.

peeras - Specifies peer's autonomous\_system number

localas - Specifies local autonomous\_system number.

maxup *number* - Specifies the maximum number of peers that will be maintained in the Up state.

Any number of group clauses can be specified containing any number of neighbor clauses. Any parameters from the neighbor clause can be specified on the group clause to provide defaults for the group.

**neighbor host** [metricout *metric*]  
[nogendefault]  
[importdefault]  
[exportdefault]  
[gateway *gateway*]  
[lcladdr *local\_address*]  
[sourcenet *network*]  
[minhello *min\_hello*]  
[minpoll *min\_poll*]  
[traceoptions *traceoptions*]

importdefault - Tells GateD to import the default route (0.0.0.0) in updates exchanged with an EGP neighbor.

exportdefault - Tells GateD to export the default route (0.0.0.0) in updates exchanged with an EGP neighbor. If neither importdefault nor exportdefault is specified, the default network is ignored when exchanging EGP updates

gateway *gateway* - If a network is not shared with the neighbor, gateway can be used to specify the next-hop gateway to use when installing routes learned from this neighbor. In this case, the default interface is the one shared with the specified gateway.

lcladdr *local\_address* - Sets the address of the local address to be used when there is a choice of interfaces. If not specified it defaults to whichever interface is shared with the neighbor.

sourcenet

*network* - Specifies the network to query in EGP Poll packets; this is normally the shared network.

minhello *min\_hello* - Specifies the minimum acceptable EGP hello interval, as a time in seconds, minutes:seconds or hours:minutes:seconds. minpoll

*min\_poll* - Specifies the minimum acceptable EGP poll interval, as a time in seconds, minutes:seconds or hours:minutes:seconds.

traceoptions - Enables tracing for a particular protocol, group, or peer.

Any number of group clauses can be specified containing any number of neighbor clauses. Any parameters from the neighbor clause can be specified on the group clause to provide defaults for the group.

---

## BGP Protocol Configuration

The section describes how to configure GateD for BGP (Border Gateway Protocol). If the BGP clause is not specified, the default is `bgp off`. The `bgp` statement is shown below:

```
bgp yes | no | on | off [ {  
    [preference preference ; ]  
    [defaultmetric metric ; ]  
    [traceoptions traceoptions ; ]  
    [group type external | internal | igp | test peeras peeras  
        [metricout metric]  
        [localas localas]  
        [nogendefault]  
        [gateway gateway]  
        [preference preference]  
        [lcladdr local_address]  
        [holdtime time]  
        [traceoptions traceoptions ]  
        [version version]  
        [passive]  
        [importdefault]  
        [exportdefault]  
        [sendbuffer bufsize]  
        [recvbuffer bufsize]  
        [spoolbuffer bufsize]  
        [keepall]  
    {  
        [allow {dest_mask ... } ; ]  
        [peer host  
            [metricout metric]  
            [localas localas]  
            [nogendefault]  
            [gateway gateway]  
            [preference preference]  
            [lcladdr local_address]  
            [holdtime time]  
            [traceoptions traceoptions]  
            [version version]
```

```

        [passive]
        [importdefault]
        [exportdefault]
        [sendbuffer bufsize]
        [rcvbuffer bufsize]
        [spoolbuffer bufsize]
        [keepall]
    ; ]
    ...
} ;
....]
} ] ;

```

### Syntax Description

<b>yes   no   on   off</b>	Turns the BGP protocol handling on or off. Default: off
<b>preference</b> <i>preference</i>	Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default is 170.
<b>defaultmetric</b>	Specifies the default metric used when propagating routes learned from other protocols. Defaults to the highest valid metric for this protocol. If a metric is not specified, the default is not to send a metric.
<b>traceoptions</b>	Enables tracing for a particular protocol, group, or peer.
<b>group</b> <i>type</i>	Lists a group of EGP peers in one autonomous system.
<b>external   internal  </b> <b>igp   test</b>	
<b>peeras</b> <i>peeras</i>	The autonomous system expected from a peer.
<b>metricout</b> <i>metric</i>	The default metric to use when sending to this peer.
<b>localas</b> <i>localas</i>	Specifies the autonomous system advertised to this peer; the default is that which has been set globally.
<b>nogendefault</b>	Specifies that this peer should not cause the automatic default to be generated.
<b>gateway</b> <i>gateway</i>	
<b>preference</b> <i>preference</i>	Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default is 170.

---

<b>locladdr</b> <i>localaddress</i>	Specifies the address to be used on the local end of the TCP connection with the peer.
<b>holdtime</b> <i>time</i>	Specifies the BGP hold time to use with this peer. Default is 180.
<b>traceoptions</b> <i>traceoptions</i>	Specifies tracing options for this peer.
<b>version</b> <i>version</i>	Specifies the version of the BGP protocol to. If not specified the highest supported version is used first and version negotiation is attempted. Default is 3.
<b>passive</b>	Specifies that active opens to this peer should not be attempted.
<b>importdefault/</b> <b>exportdefault</b>	Controls whether the default network (0.0.0.0) can be exchanged with this peer.
<b>sendbuffer</b> <i>bufsize/</i> <b>rcvbuffer</b> <i>bufsize</i>	Controls the amount of buffering asked of the kernel; the default is to configure the maximum supported, up to 65535 bytes.
<b>spoolbuffer</b> <i>bufsize</i>	Indicates that BGP should buffer data for peers when the kernel queues are full; the default is to break the connection.
<b>keepall</b>	Retains routes learned from a peer that contain one of our autonomous system numbers in their path.
<b>allow</b> <i>dest_mask</i>	
<b>peer</b>	Specifies the address of each BGP peer.
<b>neighbor</b> <i>host]</i>	
<b>metricout</b> <i>metric</i>	The metric to use. Default is 255.
<b>nogendefault</b>	
<b>importdefault</b>	Tells GateD to import the default route (0.0.0.0) in updates exchanged with an EGP neighbor.
<b>exportdefault</b>	Tells GateD to export the default route (0.0.0.0) in updates exchanged with an EGP neighbor.
<b>gateway</b> <i>gateway</i>	If a network is not shared with the neighbor, gateway can be used to specify the next-hop gateway to use when installing routes learned from this neighbor. In this case the default interface is the one shared with the specified gateway.
<b>lcladdr</b> <i>local_address</i>	Sets the address of the local address to be used when there is a choice of interfaces. If not specified it defaults to whichever interface is shared with the neighbor.
<b>sourcenet</b> <i>network</i>	Specifies the network to query in EGP Poll packets; this is normally the shared network.

**minhello** *min\_hello* Specifies the minimum acceptable EGP hello interval, as a time in seconds, minutes:seconds or hours:minutes:seconds.

**minpoll** *min\_poll* Specifies the minimum acceptable EGP poll interval, as a time in seconds, minutes:seconds or hours:minutes:seconds.

**traceoptions** Enable tracing for a particular protocol, group, or peer.  
*traceoptions*

Any number of group clauses can be specified containing any number of neighbor clauses. Any parameters from the neighbor clause can be specified on the group clause to provide defaults for the group.

BGP peers are assigned to groups based on the type and peeras, it is not possible to have two groups with the same type and peeras. Group options provide the defaults for all peers within that group.

The *local\_address* specifies the address to be used on the local end of the TCP connection with the peer. For external peers the local address must be on an interface which is shared with the peer (or for a non-local peer's configured next-hop gateway when the gateway option is used to specify this) and a session with the peer will be opened only when an interface with the appropriate local address through which the peer (gateway) address is directly reachable is operating. For other types of peers, a peer session will be maintained when any interface with the specified local address is operating. In either case, incoming connections will only be recognized as matching a configured peer if they are addressed to the configured local address.

The options *sendbuffer*, *recvbuffer*, and *spoolbuffer* are normally not needed on properly functioning systems.

## ICMP Redirect Configuration

By configuring the *redirect* statement, you control whether GateD makes routing table changes based on ICMP redirects when not functioning as a router. When functioning as a router (in other words, any interior routing protocols (RIP, HELLO, OSPF) are participating in routing on any interface), ICMP redirects are disabled. When ICMP redirects are disabled, GateD must actively remove the effects of redirects from the kernel as the kernel always processes ICMP redirects.

```
redirect yes | no | on | off [ {
    preference preference;
    interface interface_list [noredirects] ;
    trustedgateways gateway_list ;
    traceoptions traceoptions ;
} ] ;
```



---

## Syntax Description

<b>yes   no   on   off</b>	Turns ICMP redirects on or off. Default: off
<b>preference</b>	Controls the choice of routes learned via this protocol or from this autonomous system in relation to routes learned from other protocols or autonomous systems. Default is 30.
<b>interface</b> <i>interface_list</i> [noredirects]	Sets interface options on the specified interfaces. An interface list is all or a list of interface names (read the warning about interface names in the interface entry of the table in Statement Primitives), domain names, or numeric addresses. The <i>interface_list</i> option is a list of one or more interface names, wildcard names (names without a number) or addresses, or the token all, which refers to all interfaces. The noredirects option excludes the interface specified.
<b>trustedgateways</b>	Override an interface clause that disables sending or receiving protocol packets for specific peers.
<b>traceoptions</b>	Enable tracing for a particular protocol, group, or peer.

## Route Statements

Any number of static statements can be specified, each containing any number of static route definitions. The first form (the first section of the example below) defines a static route through one or more gateways. If multiple gateways are specified, they are limited by the number of multipath destinations supported (on UNIX this is almost always one). Only gateways on interfaces that are configured and up are used.

The interface list on the first form restricts static routes to a specific set of interfaces.

The second form (the second section of the example below) defines a static interface route which is used for primitive support of multiple networks on one interface.

In this example the retain parameter causes the route to be retained in the kernel after GateD is shut down. The reject parameter causes all packets to this route to be rejected. The blackhole parameter causes all packets to this route to be silently discarded.

```
static {
    dest_mask gateway gateway [gateway2 [gateway3 [...]]]
        [interface interface_list]
        [preference preference]
        [retain] [reject] [blackhole] [noinstall] ;
    ...
    dest_mask interface interface [preference preference] [retain]
        [reject] [blackhole] [noinstall] ;
    ...
} ;
```

The preference for static routes defaults to 60.

---

**Note** The reject and blackhole parameters are not supported by all systems. The noinstall parameter prevents this route from being installed in the kernel.

---

## Control Statements

Control statements define routes that are imported from routing peers and routes that are exported to these peers. This section describes the use of the import and export statements.

### Import Statements

An `import_list` consists of zero or more destinations (with optional mask). One of two parameters may be specified; `restrict` to prevent a set of destinations from being imported, or a specific preference for this set of destinations.

`dest_mask [[restrict] | [preference preference]];`

The contents of an `import_list` are sorted internally so that entries with the most specific masks are examined first. The order in which `dest_mask` entries are specified does not matter.

If no import list is specified, all routes will be accepted. If an import list is specified, the import list is scanned for a match. If no match is found, the route is discarded. Rephrased, an all `restrict` entry is assumed in an import list.

```
import proto bgp|egp as autonomous system restrict ;
import proto bgp|egp as autonomous system
    [preference preference] {
import_list
    } ;
import proto bgp aspath aspath_spec restrict ;
import proto bgp aspath aspath_spec
    [preference preference] {
import_list
    } ;
import proto rip|hello|redirect restrict ;
import proto rip|hello|redirect
    [preference preference] {
import_list
    } ;
import proto rip|hello|redirect interface interface_list restrict ;
import proto rip|hello|redirect interface interface_list
    [preference preference] {
import_list
    } ;
import proto rip|hello|redirect gateway gateway_list restrict ;
import proto rip|hello|redirect gateway gateway_list
    [preference preference] {
import_list
    } ;
import proto ospfase [tag ospf_tag] restrict ;
import proto ospfase [tag ospf_tag]
    [preference preference] [{
import_list
    }] ;
```

If an OSPF type is specified, only routes of that type will be considered for import, otherwise either type will be considered. If an `ospf_tag` specification is given, only routes matching that tag specification will be considered, otherwise any tag will be considered. An OSPF tag specification is a decimal, hexadecimal or dotted quad number.

If more than one import statement relevant to a protocol is specified, they are processed most specific to least specific (in other words, for RIP and HELLO, gateway, interface, and protocol), then in the order specified in the configuration file.

---

## Export Statements

The `export_list` specifies exportation based on the origin of a route to a destination:

```
export proto bgp|egp as autonomous system restrict ;
export proto bgp|egp as autonomous system
    [metric metric] {
    export_list
} ;
export proto rip|hello restrict ;
export proto rip|hello [metric metric] {
    export_list
} ;
export proto rip|hello interface interface_list restrict ;
export proto rip|hello interface interface_list
    [metric metric] {
    export_list
} ;
export proto rip|hello gateway gateway_list restrict ;
export proto rip|hello gateway gateway_list
    [metric metric] {
    export_list
} ;

export proto ospfase [type 1|2] [tag ospf_tag] restrict ;
export proto ospfase [type 1|2] [tag ospf_tag]
    [cost ospf_cost] {
    export_list
} ;
```

The following export statements demonstrate the use of a tag to specify particular routes.

```
proto bgp|egp as autonomous system
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
proto rip|hello|direct|static|default
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
proto rip|hello|direct|static|default interface interface_list
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
proto rip|hello gateway gateway_list
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
proto ospf [restrict] | [metric metric] [ {
    announce_list ;
} ] ;
proto ospfase [restrict | metric metric] [ {
    announce_list ;
} ] ;
proto proto aspath aspath_spec
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
proto proto tag tag
    [restrict] | [metric metric] [ {
    announce_list
} ] ;
```

If a tag is specified, only routes with that tag will be considered, otherwise any tag will be considered. An OSPF tag on an export statement may be a decimal, hexadecimal, or AS to generate a tag based on the AS path of route being announced. An OSPF tag on an export list is just a 31 bit number that is matched against the tag present (if any) on that route.

If more than one export statement relevant to a protocol is specified, they are processed most specific to least specific (in other words, for RIP and HELLO, gateway, interface and protocol), then in the order specified in the configuration file.

By default, interface routes are exported to all protocols. RIP and HELLO also export their own routes. An export specification with just a restrict will prevent these defaults from being exported.

---

**Note** It is not possible to change the metric RIP and HELLO use for their own routes; any attempt to override it will be silently ignored.

---

Any protocol may be specified for import lists referring to aspaths and tags. AS paths are most meaningful with BGP and OSPF ASE routes, but are generated for all routes. tags are currently only meaningful for OSPF ASE routes.

An announce\_list consists of zero or more destinations (with optional mask). One of two parameters may be specified, restrict to prevent a set of destinations from being exported or a specific metric for this set of destinations.

```
dest_mask [[restrict] | [metric metric]];
```

Note that the contents of an announce\_list are sorted internally so that entries with the most specific masks are examined first. The order in which dest\_mask entries are specified does not matter.

If no announce\_list is specified, all destinations are announced. If an announce list is specified, an all restrict is assumed. Therefore, an empty announce list is the equivalent of all restrict.

---

**Note** To announce routes which specify a next hop of the loopback interface (in other words, static and internally generated default routes) via RIP or HELLO it is necessary to specify the metric at some level in the propagate clause. Just setting a default metric for RIP or HELLO is not sufficient.

---

An AS path specification (aspath\_spec) is used to match one or more AS paths.

```
aspath regexp origin [igp|egp|incomplete|any]
```

where the *regexp* is a regular expression over the set of AS numbers as defined in RFC-1164 section 4.2.