



Setting Up Your GSS from the CLI

This chapter describes how to configure your GSS devices from the CLI and connect it to the GSS network. This process describes how to configure the GSS as a primary GSSM, a standby GSSM, or as a GSS device.

This chapter contains the following sections:

- [Initial Setup Quick Start](#)
- [Logging in to the CLI and Enabling Privileged EXEC Mode](#)
- [Setting the System Clock](#)
- [Configuring a Hostname for the GSS Device](#)
- [Configuring an Ethernet Interface on a GSS Device](#)
- [Specifying Name Servers](#)
- [Configuring an IP Route for the GSS](#)
- [Resolving a Host or Domain Name to an IP Address](#)
- [Configuring a Primary GSSM](#)
- [Configuring a Standby GSSM](#)
- [Configuring a Global Site Selector](#)
- [Where to Go Next](#)

Initial Setup Quick Start

Table 4-1 is a quick start configuration table designed to help you configure your GSS quickly from the CLI. This table provides information and examples on the following basic steps to perform the following:

- Configuring the system clock for the GSS device
- Specifying a qualified hostname for the GSS device
- Configuring Ethernet 0 and Ethernet 1
- Configuring a default gateway
- Entering the IP addresses of the name servers (maximum of eight)
- Configuring the primary GSSM, standby GSSM, and GSS devices that comprise your GSS network

Table 4-1 Initial Setup Quick Start

Task and Command Example

1. If you have not already done so, power on and boot the GSS (as described in the *Cisco Global Site Selector Hardware Installation Guide*).
2. If you have not already done so, enable a remote access protocol (such as Telnet or SSH) to access the GSS CLI. See [Chapter 3, Accessing the GSS CLI](#).
3. Log on to the CLI, and at the GSS CLI prompt, enable privileged EXEC mode.

```
localhost.localdomain> enable
```

By default, the hostname for GSS devices is localhost.localdomain. This name changes once you configure the hostname for the device.

4. Enable privileged EXEC mode.

```
localhost.localdomain> enable
```

5. Configure the time using the **clock set** command. Enter the time in the *hh:mm:ss* format and the date in *month dd yyyy* format.

```
localhost.localdomain# clock set 12:10:05 Feb 15 2006
```

Table 4-1 Initial Setup Quick Start (continued)

Task and Command Example

6. (Optional) If you intend to use an NTP server to synchronize the GSS system clock, access global configuration mode and specify the NTP server.

```
localhost.localdomain# config  
localhost.localdomain(config)# ntp-server 172.16.1.2 172.16.1.3  
localhost.localdomain(config)# ntp enable
```

7. Configure a hostname for the GSS device. The **hostname** command requires a fully qualified hostname, which requires at least one period (.) in the name.

```
localhost.localdomain(config)# hostname gssm1.example.com
```

8. From global configuration mode, enter interface configuration mode and configure the attributes of GSS interface Ethernet 0 or Ethernet 1. Each GSS device contains two Ethernet interfaces, 0 and 1.

```
gssm1.cisco.com(config)# interface ethernet 0  
gssm1.cisco.com(config-eth0)# speed 100  
gssm1.cisco.com(config-eth0)# duplex full
```

You cannot execute interface commands while the GSS software is running (for example, serving DNS requests). You must enter the **gss stop** command to stop the GSS software before executing the **interface ethernet** command.

9. Use the **gss-communications** command to configure a GSS Ethernet interface as the designated network interface for GSS device communication.

```
gssm1.cisco.com(config-eth0)# gss-communications  
gssm1.cisco.com(config-eth0)# exit  
gssm1.cisco.com(config)#
```

10. Configure the IP address and subnet mask for the interface.

```
gssm1.cisco.com(config-eth0)# ip address 192.168.3.24  
255.255.255.0
```

11. Use the **gss-tcp-keepalives** command to designate either Ethernet 0 or Ethernet 1 for TCP and HTTP HEAD keepalive communication.

```
gssm1.cisco.com(config)# interface eth1  
gssm1.cisco.com(config-eth1)# gss-tcp-keepalives
```

Table 4-1 Initial Setup Quick Start (continued)

Task and Command Example
<p>12. Exit interface configuration mode.</p> <pre>gssm1.cisco.com(config-eth1)# exit gssm1.cisco.coml(config)#</pre>
<p>13. Define a default gateway for the GSS device.</p> <pre>gssm1.cisco.com(config)# ip default-gateway 172.16.7.18</pre>
<p>14. Configure the domain name server or servers to be used by the GSS device. You can enter individual IP addresses or specify a maximum of eight name servers using a comma-separated or space-separated list.</p> <pre>gssm1.cisco.com(config)# ip name-server 192.168.12.1, 192.168.12.5, 192.168.12.7</pre>
<p>15. Exit global configuration mode.</p> <pre>gssm1.cisco.com(config)#exit</pre>
<p>16. Configure the primary GSSM in your GSS network. You must have a primary GSSM configured and enabled before you can enable a standby GSSM and GSS devices.</p> <pre>gssm1.cisco.com# gss enable gssm-primary</pre>
<p>17. Configure the standby (backup) GSSM in your GSS network and associate it with the DNS name or IP address of the primary GSSM. The standby GSSM is intended to be a backup device to be used on a temporary basis until the primary GSSM comes back online.</p> <pre>gssm2.example.com# gss enable gssm-standby gssm1.example.com</pre>
<p>18. Enable each GSS device as a GSS and direct it to the primary GSSM in your GSS network.</p> <pre>gss1.example.com# gss enable gss gssm1.example.com</pre>
<p>19. Save your configuration changes to memory.</p> <pre>gssm1.example.com# copy running-config startup-config</pre>

Logging in to the CLI and Enabling Privileged EXEC Mode

To log in to a GSS device and enable privileged EXEC mode at the CLI, perform the following steps:

1. Press the power control button on the GSS. After the GSS boot process completes, the software prompts you to log in to the device.
2. If you are remotely logging in to the GSS device (Global Site Selector or Global Site Selector Manager) through Telnet or SSH, enter the hostname or IP address of the GSS to access the CLI.

Otherwise, if you are using a direct serial connection between your terminal and the GSS device, use a terminal emulation program to access the GSS CLI. For details about making a direct connection to the GSS device using a dedicated terminal and about establishing a remote connection using SSH or Telnet, see [Chapter 3, Accessing the GSS CLI](#).

3. Specify your GSS administrative username and password to log in to the GSS device. The CLI prompt appears.

```
localhost.localdomain>
```

4. At the CLI prompt, enable privileged EXEC mode.

```
localhost.localdomain> enable  
localhost.localdomain#
```

The prompt changes from the user-level EXEC right angle bracket (>) prompt to the privileged-level EXEC pound sign (#).

Setting the System Clock

To set the date, time, or time zone for a GSS device, use the **clock** command. When you enter this command, the GSS device displays the current date and time.

This section contains the following topics:

- [Setting the Time and Date](#)
- [Setting the Time Zone](#)
- [Setting User-Defined Time Zones](#)
- [Setting User-Defined Daylight Saving Time](#)
- [Setting Daylight Saving Time to Another Predefined Summer Time](#)
- [Setting the Hardware Clock](#)
- [Showing the Date, Time, and Timezone](#)

Setting the Time and Date

Use the **clock set** command to set the time and the date for a GSS device. Enter the time and date:

- **Time**—Hour, minutes, and seconds as integers in military-time (24-hour) format, separated by colons.
- **Date**—Enter the month, day, and year as integers with colon (:) characters separating them.

The syntax for the **clock set** command is as follows:

```
clock set hh:mm:ss MONTH DD YYYY
```

The keywords and arguments are as follows:

- **set**—Sets the device clock to the date and time provided.
- *hh:mm:ss*—Current time to which the GSS device clock is being reset. Specify one or two digits for the hours, minutes, and seconds in military-time (24-hour) format, separated by colons.

- *MONTH DD YYYY*—Current date to which the GSS device clock is being reset. Specify the full name of the month, one or two digits for the day, and four digits for the year. The following month names are recognized: January, February, March, April, May, June, July, August, September, October, November, and December.

For example, to specify a time of 12:10 and a date of February 15, 2006, enter:

```
localhost.localdomain# clock set 12:10:05 February 15 2006
```



Note

If you previously enabled NTP on a GSS, the GSS prevents you from using the **clock set** command and displays an error message. If you want to manually set the clock for the GSS, first disable NTP using the **no ntp enable** command before setting the clock. See the “[Setting the Hardware Clock](#)” section for more information.

Setting the Time Zone

The time stored in the GSS is the local time. Use the **clock timezone** command to specify a time zone for the GSS, synchronizing the log timestamps to a new timezone. The name of the timezone. Enter **?** to list all supported time zones, countries, continents, and cities.

There are a number of options available to set the local time zone for your GSS:

- Standard time zone (for example, GMT, EST, UTC).
- Country or part of a continent (for example, America, Europe, Egypt)
- Specific city (for example, New York, Paris)

The syntax for this command is as follows:

```
clock timezone timezonename
```

The *timezonename* is the name of the time zone you specify for the GSS. Enter **?** to list all supported time zone names.

For example, to specify the Greenwich Mean Time (GMT) time zone, enter:

```
localhost.localdomain# clock timezone GMT
```

For example, to specify the time zone to the local time in Paris, enter:

```
localhost.localdomain# clock timezone europe paris
```

Setting User-Defined Time Zones

The **clock user-timezone** command allows you to set user-defined time zones.

The syntax for this command is as follows:

```
clock user-timezone timezonename | hour_offset <-23,+24> | minute_offset <0-59>
```

The keywords and arguments are as follows:

- *timezonename*—Name of the user-defined time zone.
- *hour_offset*—Hour offset for the user-defined time zone. Valid values range from -23 to +24.
- *minute_offset*—Minute offset for the user-defined time zone. Valid values range from 0 to 59.

Setting User-Defined Daylight Saving Time

The **clock user-summer-time** command allows you to set a user-defined daylight saving time.

The syntax for this command is as follows:

```
clock user-summer-time summer-time name | start time (hh:mm) <0-23> | start day <Friday, Monday, Saturday, Sunday, Thursday, Tuesday, Wednesday> | start week <1-5> | start month <April, August, December, February, January, July, June, March, May, November, October, September> | end time (hh:mm) <0-23> | end day <Friday, Monday, Saturday, Sunday, Thursday, Tuesday, Wednesday> | end week <1-5> | end month <April, August, December, February, January, July, June, March, May, November, October, September> | offset (in minutes) <0-1440>
```

The keywords and arguments are as follows:

- *summer-time name*—Name of the user-defined summer time.
- *start time*—Start time for the user-defined summer time in hours and minutes. Valid values are from 0-23.
- *start day*—Start day for the user-defined summer time. Valid values are Friday, Monday, Saturday, Sunday, Thursday, Tuesday, and Wednesday.

- *start week*—Start week for the user-defined summer time. Valid values are from 1-5.
- *start month*—Start month for the user-defined summer time. Valid values are April, August, December, February, January, July, June, March, May, November, October, and September.
- *end time*—End time for the user-defined summer time in hours and minutes. Valid values are from 0-23.
- *end day*—End day for the user-defined summer time. Valid values are Friday, Monday, Saturday, Sunday, Thursday, Tuesday, and Wednesday.
- *end week*—End week for the user-defined summer time. Valid values are from 1-5.
- *end month*—End month for the user-defined summer time. Valid values are April, August, December, February, January, July, June, March, May, November, October, and September.
- *offset*—Offset (in minutes) for the user-defined time zone. Valid values are from 0-1440.

Setting Daylight Saving Time to Another Predefined Summer Time

The **clock summer-time** command allows you to set daylight saving time to another predefined summer time.

The syntax for this command is as follows:

```
clock summer-time timezone <ADT (Atlantic Daylight Time) | AKDT  
(Alaska Standard Daylight Time) | CDT (Central Daylight Time) | EDT  
(Eastern Daylight Time) | MDT (Mountain Daylight Time) | PDT (Pacific  
Daylight Time)
```

The keywords and arguments are as follows:

- **summer-time**—Allows you to set daylight saving time to another predefined summer time.
- *timezonename*—Name of the other predefined summer time zone. Valid values are ADT (*Atlantic Daylight Time*), AKDT (*Alaska Standard Daylight Time*), CDT (*Central Daylight Time*), EDT (*Eastern Daylight Time*), MDT (*Mountain Daylight Time*), and PDT (*Pacific Daylight Time*)

For example, if you wish to set up standard time and daylight saving time in the Eastern Time Zone (that is, Eastern Standard Time (EST) Universal Time Coordinated (UTC -5)) and its daylight saving time (that is, Eastern Daylight Time (EDT) (UTC -4), use the **clock user-timezone**, **summer-time**, and **user-summer-time** commands as shown in [Table 4-2](#).

EDT begins at 2:00 a.m. local time on the first Sunday in April and ends at 2:00 a.m. local time on the last Sunday of October.

Table 4-2 Using the Clock Commands—Example

1. Set the clock based on UTC.

```
gss2.gss-tb6.com# clock set 7:45:00 September 7 2006
gss2.gss-tb6.com# show clock
System time: Thu Sep 7 07:45:05 UTC 2006.
```

2. Set the clock to EST by using the **user-timezone** command.

```
Timezone set.com# clock user-timezone EST -5 0
Please restart the GSS (reload) to sync log timestamps to new
timezone.
gss2.gss-tb6.com# show clock
System time: Thu Sep 7 02:45:17 EST 2006
```



Note

You need to set the **user-timezone** before using the **summer-time** command.

I

3. Set the summer time on the GSS in one of two ways:**a. Enter the `summer-time` command.**

```
gss1.gss-tb6.com#clock summer-time ?
  ADT  Atlantic Daylight Time: 2 am 1st Sunday April - 2 am last
        Sunday Oct, +60 min
  AKDT Alaska Standard Daylight Time: 2 am 1st Sunday April - 2
        am last Sunday
        Oct, +60 min
  CDT  Central Daylight Time: 2 am 1st Sunday April - 2 am last
        Sunday Oct, +60 min
  EDT  Eastern Daylight Time: 2 am 1st Sunday April - 2 am last
        Sunday Oct
  MDT  Mountain Daylight Time: 2 am 1st Sunday April - 2 am last
        Sunday Oct, +60 min
  PDT  Pacific Daylight Time: 2 am 1st Sunday April - 2 am last
        Sunday Oct, +60 min
gss1.gss-tb6.com#clock summer-time EDT
Timezone set.
Please restart the GSS (reload) to sync log timestamps to new
timezone.
```

b. Enter the `user-summer-time` command.

```
gss2.gss-tb6.com# clock user-summer-time EDT 2:00 Sunday 1 April
2:00 Sunday 5 October 60
Timezone set.
Please restart the GSS (reload) to sync log timestamps to new
timezone.
gss2.gss-tb6.com# show clock
System time: Thu Sep 7 03:47:07 EDT 2006
```

If you reside in one of the following timezones, ADT, AKDT, CDT, EDT, MDT, or PDT, you use the **clock summer-time** command as shown in step 3a to set the summer-time on the GSS. If you need to configure a custom summer-time, use the **clock user-summer-time** command and follow step 3b instead.

In this example, the summer-times are predefined summer times. After it is finished, the clock will be set back to the user-timezone after the daylight saving time duration is covered. In this particular instance, the time zone will be set back to EST after 2:00 a.m. on the last Sunday in October.

**Note**

You set both the **summer-time** and **user-summer-time** command options with respect to UTC only, and not the **user-timezone**. Also, there is no relation to the **clock timezone** command that was used to establish a time zone on the GSS.

Setting the Hardware Clock

The hardware clock is powered by a Lithium battery on the motherboard of the GSS. The system clock is a software concept, rather than an actual physical entity. It is updated by the Network Time Protocol (NTP) or by the **clock set** command. For more information on NTP, see the [“Synchronizing the GSS System Clock with an NTP Server”](#) section, while the [“Setting the Time and Date”](#) section contains more information on the **clock set** command.

The **clock update-calendar** and **clock read-calendar** commands provide a way for you to synchronize the hardware clock and the system clock without having to reload the GSS. You use **clock update-calendar** to update the hardware calendar from the system clock and **clock read-calendar** to read the hardware calendar into the system clock.

Specifically, you use **clock update-calendar** when the system clock is set via NTP and you wish to synchronize the system time with the hardware clock. You then use **clock read-calendar** when the system clock is set via NTP and you wish to revert back to using the hardware clock.

The syntax for these commands is as follows:

clock update-calendar

clock read-calendar

For example, to update the hardware calendar from the system clock, enter:

```
localhost.localdomain# clock update-calendar
```

For example, to read the hardware calendar into the system clock, enter:

```
localhost.localdomain# clock read-calendar
```

Synchronizing the GSS System Clock with an NTP Server

NTP enables you to synchronize the GSS system clock to a time server. NTP is a protocol designed to synchronize the clocks of computers over a network. NTP assures accurate local time-keeping with references to radio and atomic clocks. The NTP protocol is capable of synchronizing distributed clocks within milliseconds over long time periods. You can specify a maximum of four NTP servers.



Note

If you are using DNS sticky and your network contains multiple GSS devices operating in a global sticky mesh, we strongly recommend that you first synchronize the system clock of each GSS device in the mesh with an NTP server.

Use the **ntp-server** global configuration mode command to specify one or more NTP servers for GSS clock synchronization. To disable an NTP time server, use the **no** form of this command. The syntax for this command is as follows:

```
ntp-server ip_or_host
```

The *ip_or_host* argument specifies the IP address or hostname of the NTP public time server that provides the clock synchronization. You can specify a maximum of four IP addresses or hostnames. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic hostname (for example, myhost.mydomain.com).

Use the **ntp enable** global configuration mode command to enable the NTP service. To disable NTP, use the **no** form of this command. The syntax of this command is:

```
ntp enable
```

For example, to specify the IP addresses of two NTP time servers, enter:

```
localhost.localdomain# config  
localhost.localdomain(config)# ntp-server 172.16.1.2 172.16.1.3  
localhost.localdomain(config)# ntp enable
```

To view if NTP is enabled for the GSS device, enter:

```
localhost.localdomain# show ntp
Ntp is enabled
```

```
ntp-server 172.16.1.2
ntp-server 172.16.1.3
```

To remove a specified NTP server, enter:

```
localhost.localdomain(config)# no ntp-server 172.16.1.3
```

To disable NTP, enter:

```
localhost.localdomain(config)# no ntp enable
```

Showing the Date, Time, and Timezone

Use the **show clock** command to display the current date, time, and timezone name.

```
localhost.localdomain# show clock
System time: Wed February 15 20:55:36 UTC 2006
```

[Table 4-3](#) describes the fields in the **show clock** command output.

Table 4-3 Field Descriptions for the show clock Command

Field	Description
Date	Current date in the format of day, month, and year.
Time	Current time in the format of hour, minute, and second, for example, 16:23:45.
Timezone	Name of the configured time zone.
Year	Current year.

Configuring a Hostname for the GSS Device

By default, the hostname for GSS devices is localhost.localdomain. The hostname is used for the command prompts and default configuration filenames. To configure a qualified hostname for the GSS device, use the **hostname** command. This name changes once you configure the hostname for the device.

The **hostname** command requires a fully qualified hostname, which requires at least one period (.) in the name (for example, hostname.foo.com). The **no** form of this command erases the configured hostname and restores the default value.

When you specify a hostname for a GSS (primary GSSM, standby GSSM, or GSS device) that is operating in a lab network environment, the top-level domain of the hostname cannot begin with a numerical value. For example, you cannot name a primary GSSM as gssm.1lab. If you attempt to create or change a hostname for a top-level domain to a name that begins with a number, the following message appears:

```
Top level domains of hostnames cannot begin with a number
```

For the purposes of GSS interdevice communications, configure the hostname on the same interface (eth0 or eth1) that is being used for GSS communications, as set using the **gss-communications** command.

The syntax for this global configuration mode command is:

hostname *host_name*

Specify the new hostname for the GSS device as a case sensitive text string that contains from 1 to 22 alphanumeric characters.

For example, to change the hostname to *gssm1.cisco.com*, enter:

```
localhost.localdomain(config)# hostname gssm1.cisco.com  
gssm1.cisco.com(config)#
```

To remove the hostname and set it to the default localhost.localdomain, enter:

```
gssm1.cisco.com(config)# no hostname gssm1.cisco.com  
localhost.localdomain(config)#
```

Configuring an Ethernet Interface on a GSS Device

Your GSS comes with one integrated dual-port Ethernet controller. This controller provides an interface for connecting to 10-Mbps, 100-Mbps, or 1000-Mbps networks and supports autonegotiate, full-duplex, or half-duplex operations on an Ethernet LAN.

To configure a GSS Ethernet interface, access global configuration mode and use the **interface ethernet** command. The syntax for this command is as follows:

```
interface ethernet {0 | 1} {autosense | duplex {auto | full | half} |  
  gss-communications | gss-tcp-keepalives | ip address {ip-address  
  netmask} | no | shutdown | speed {mbits | auto}
```

If desired, you can use the following CLI commands to configure specific Ethernet interface settings:

- **autosense**
- **duplex** {**auto** | **full** | **half**}
- **gss-communications**
- **gss-tcp-keepalives**
- **ip address** {*ip-address netmask*}
- **shutdown**
- **speed** {*mbits* | **auto**}

The following sections provide detailed information on:

- [Configuring an Interface](#)
- [Configuring Autosense](#)
- [Configuring Interface Duplex Operation](#)
- [Configuring Interface Speed](#)
- [Configuring GSS Interdevice Communication](#)
- [Configuring an Interface for TCP and HTTP HEAD Keepalive Communication](#)
- [Setting the IP Address and Subnet Mask of the Ethernet Interface](#)
- [Shutting Down an Interface](#)
- [Showing Interface Information](#)
- [Outputting a Record of TCP Traffic](#)

Configuring an Interface

Use the **interface ethernet** command to configure an Ethernet interface on a GSS device. The syntax for entering an Ethernet interface is as follows:

```
interface ethernet {0 | 1}
```

The options are:

- **0**—Specifies the first Ethernet interface on a GSS device
- **1**—Specifies the second Ethernet interface on a GSS device

For example, to configure Ethernet interface port 0 on a GSS and access the interface mode, enter:

```
gssm1.cisco.com(config)# interface ethernet 0
```

The GSS changes from configuration mode to the specific interface mode.

```
gssm1.cisco.com(config-eth0)#
```

Configuring Autosense

The **autosense** command enables the current GSS interface to select the proper duplex mode (for example, full duplex, half duplex) for communicating with other network devices. The GSS automatically detects the network line speed (Fast Ethernet only) and duplex of incoming signals, and it synchronizes those parameters during data transfer. Autonegotiation enables the GSS and the other devices on the link to achieve the maximum common level of operation. Autosense is enabled by default.



Note

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **autosense** command.

To configure autosense for interface Ethernet 0, enter:

```
gssml.cisco.com(config)# interface eth0
```

or

```
gssml.cisco.com(config)# interface eth0  
gssml.cisco.com(config-eth0)# autosense
```

When **autosense** is on, manual configurations are overridden. To prevent your configuration from being overwritten, disable **autosense** before configuring an Ethernet interface.

To disable autosense, use the **no** form of this command. For example, enter:

```
gssml.cisco.com(config-eth0)# no autosense
```

Configuring Interface Duplex Operation

The **duplex** command enables you to configure an Ethernet interface for full- or half- duplex operation. Full duplex allows data to travel in both directions at the same time through an Ethernet interface. A half-duplex setting ensures that data travels only in one direction at any given time. Although full duplex is faster, the Ethernet interfaces sometimes cannot operate effectively in this mode. If you encounter excessive collisions or network errors, configure the interface for half duplex rather than full duplex. To disable duplex operation, use the **no** form of the command.



Note

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **duplex** command.

The syntax is as follows:

```
interface ethernet {0 | 1} duplex {auto | full | half}
```

The options are:

- **0**—Specifies the first Ethernet interface on a GSS device
- **1**—Specifies the second Ethernet interface on a GSS device
- **auto**—Resets the Fast Ethernet and Gigabit Ethernet ports to automatically negotiate port speed and duplex of incoming signals.
- **full**—Configures an interface for full-duplex operation., which allows data to travel in both directions at the same time.
- **half**—Configures an interface for half-duplex operation, which ensures that data travels in one direction only at any given time.

**Note**

When the GSS 4491 is forced to 1000-Mbps full duplex through the CLI, it goes into autonegotiate mode but operates as specified by advertising only “1000-full.” When the GSS 4491 is forced to any other speed or duplex setting, it advertises “forced” rather than “negotiated.”

Specify an interface bandwidth (Mbps) using the **speed** command before you configure full- or half-duplex. If you enter the **duplex full or duplex half** command without specifying an interface bandwidth, the following error message appears:

```
Duplex will not be set until speed is set to a non-auto value
```

To configure full duplex for interface Ethernet 0, enter:

```
gssm1.cisco.com(config)# interface eth0 duplex full
```

or

```
gssm1.cisco.com(config)# interface eth0  
gssm1.cisco.com(config-eth0)# duplex full
```

To disable duplex operation for interface Ethernet 0, enter:

```
gssm1.cisco.com(config-eth0)# no duplex
```

Configuring Interface Speed

The **speed** option sets the bandwidth on Fast Ethernet interfaces only. Gigabit Ethernet interfaces run at 1000 Mbps only and are not user-configurable. To restore default values, use the **no** form of this command.



Note

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **speed** command.

The syntax is:

```
interface ethernet {0 | 1} speed mbits
```

Specify the bandwidth size in megabits per second (Mbps). The default speed for a GSS interface is autonegotiate. The available ranges are as follows:

- **10**—Initiates 10-Mbps operation
- **100**—Initiates 100-Mbps operation
- **1000**—Initiates 1000-Mbps operation
- **auto**—Enables the GSS to autonegotiate with other devices (default)



Note

The interface speed of the GSS 4490 cannot be configured to 1000-Mbps by using the **interface ethernet {0 | 1} speed** command. If you attempt to specify an operating speed of 1000, the GSS 4490 remains set at the previous setting (as displayed through the **show interface** command). To enable a GSS 4490 interface to operate at 1000 Mbps, specify **auto**. The autonegotiate selection allows the GSS 4490 autonegotiate to 1000 Mbps with other devices.

To set the bandwidth on Ethernet 0, enter:

```
gssm1.cisco.com(config)# interface eth0 speed 100
```

or

```
gssm1.cisco.com(config)# interface eth0  
gssm1.cisco.com(config-eth0)# speed 100
```

To restore the default setting of autonegotiate for interface Ethernet 0, enter:

```
gssm1.cisco.com(config-eth0)# no speed
```

Configuring GSS Interdevice Communication

During inter-GSS communication, all GSS devices listen for configuration and status updates on only one interface. The default inter-GSS communication interface is Ethernet 0. To designate Ethernet interface 1 for inter-GSS communication, use the **gss-communications** option.

**Note**

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **gss-communications** command.

To designate Ethernet 1 for GSS interdevice communication, enter:

```
gssml.cisco.com(config)# interface eth1 gss-communications
```

or

```
gssml.cisco.com(config)# interface eth1  
gssml.cisco.com(config-eth1)# gss-communications
```

Configuring an Interface for TCP and HTTP HEAD Keepalive Communication

To designate one of the two GSS Ethernet interfaces as the source for TCP and HTTP HEAD keepalive communication, use the **gss-tcp-keepalives** option. Only one Ethernet interface (0 or 1) can be designated for TCP and HTTP HEAD keepalive communication.

**Note**

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **gss-tcp-keepalives** command.

For example, to designate Ethernet 1 for TCP and HTTP HEAD keepalive communication, enter:

```
gssml.cisco.com(config)# interface eth1 gss-tcp-keepalives
```

or

```
gssml.cisco.com(config)# interface eth1  
gssml.cisco.com(config-eth1)# gss-tcp-keepalives
```

Setting the IP Address and Subnet Mask of the Ethernet Interface

Use the **ip address** command to assign an IP address and subnet mask to an Ethernet interface. You cannot assign the same IP address to more than one interface. To disable a specific IP address, use the **no** form of the command.

The syntax is:

```
ip address ip-address ip-subnet
```

The variables are:

- *ip-address*—IP address of the Ethernet interface. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *ip-subnet*—Subnet mask of the interface. The subnet mask of the interface in dotted-decimal notation (for example, 255.255.255.0).

**Note**

You cannot enter interface commands while the GSS software is running (for example, serving DNS requests). Enter the **gss stop** command to stop the GSS software before executing the **ip address** command.

To assign an IP address to Ethernet 0, enter:

```
gssm1.cisco.com(config)# interface eth0 ip address 192.168.10.2  
255.255.255.0
```

or

```
gssm1.cisco.com(config)# interface eth0  
gssm1.cisco.com(config-eth0)# ip address 192.168.10.2 255.255.255.0
```

To remove an IP address and subnet mask for interface Ethernet 0, enter:

```
gssm1.cisco.com(config)# interface eth0  
gssm1.cisco.com(config-eth0)# no ip address
```

Shutting Down an Interface

Use the **shutdown** command in interface configuration mode to shut down a particular Ethernet interface on the GSS device.

To shut down interface Ethernet 1, enter:

```
gssm1.cisco.com(config)# interface eth1 shutdown
```

or

```
gssm1.cisco.com(config)# interface eth1  
gssm1.cisco.com(config-eth1)# shutdown
```

Showing Interface Information

To display GSS hardware interface information for Ethernet interface 0 or 1, including interface statistics, use the **show interface** command.

```
show interface {eth0 | eth1}
```

For example, to display information for Ethernet interface 0, enter:

```
gssml.cisco.com# show interface eth0
Interface eth0
  ip address 10.86.209.167 255.255.254.0
  gss-communications

Interface State
  Link is up
  negotiated, 100 mbps, full duplex
  Supported modes: 10-half, 10-full, 100-half, 100-full, 1000-full
  Advertised modes: 10-half, 10-full, 100-half, 100-full, 1000-full

Interface statistics
  eth0      Link encap:Ethernet  HWaddr 00:C0:9F:35:D1:64
            inet addr:10.86.209.167  Bcast:10.86.209.255
Mask:255.255.254.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:583003 errors:0 dropped:0 overruns:0 frame:0
            TX packets:114048 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:45735671 (43.6 Mb)  TX bytes:9810113 (9.3 Mb)
            Base address:0xbc00 Memory:fc9e0000-fca00000
```

Outputting a Record of TCP Traffic

Use the **tcpdump interface** command to display a record at the CLI of all TCP traffic transmitted from and received by an Ethernet interface. GSS traffic information continuously displays on screen until you press **Ctrl-C** to cancel the operation. The **protocol**, **port**, **network**, and **file** options of the **tcpdump** command allow you to filter traffic and capture only the traffic of certain protocols, going to or coming from certain hosts or certain ports.

The syntax for this command is as follows:

```
tcpdump interface { any | eth0 | eth1 } | protocol { any | icmp | tcp | udp } | host
  { any | ip_or_host } | port { any | port } | network { any | ip-address
  ip-subnet } | file { filename }
```

The keywords and arguments are as follows:

- **any**—Instructs the GSS software to accept all selections for an associated option. For example, if you enter **tcpdump interface any any**, the GSS filters the ICMP, TCP, and UDP IP protocols on Ethernet 0 and 1.
- **eth0**—Outputs a record of all traffic transmitted from and received by interface Ethernet 0.
- **eth1**—Outputs a record of all traffic transmitted from and received by Ethernet 1.
- **protocol {icmpltcp|udp}**—Filters the protocol for the traffic type. Recognized IP protocols include:
 - **icmp**—Internet Control Message Protocol
 - **tcp**—Transmission Control Protocol
 - **udp**—User Datagram Protocol
- **host {ip_or_host}**—Filters the host machine that is the source or destination of the packet. The software uses the IP address or hostname of the device that is the source or destination of the packet.
- **port {port}**—Filters the source or destination port of the packet.
- **network {ip-address ip-subnet}**—Filters the network IP address from which the packet originated. The software uses the *ip-address* and *ip-subnet* arguments to match the incoming packet to a source network.
- **file {filename}**—Enables you to capture raw data to a file. Then you can open the captured raw data in a Sniffer tool. When capturing data to a file, the entire packet is captured. A maximum of 20,000 filtered packets can be captured to disk. This packet limit is meant to prevent you from accidentally filling up the disk when capturing data using the **tcpdump** command.

If the file parameter is not specified, captured data is dumped to the screen. In that case, only header data is displayed and there is no limit to number of packets captured.

If you execute the **tcpdump** command without any specified options, no filtering is performed. If you want to use the defaults for the remaining **tcpdump** command parameters, press **Enter** at each option. No further filtering is performed by the GSS, other than what has been specified. For example, if you enter **tcpdump interface eth0 protocol tcp**, the GSS performs only IP protocol filtering and does not perform host, port, or network filtering.

The following is an example of the **tcpdump interface** command and its output:

```
gssm1.cisco.com# tcpdump interface eth0
Kernel filter, protocol ALL, datagram packet socket
tcpdump: listening on eth0
19:20:45.678641 > gssm.cisco.com.ssh > 10.1.2.3.1178: P
2126255246:2126255346(100) ack 4828790 win 32680 (DF) [tos 0x10]
19:20:45.680534 > gssm.cisco.com.49165 > gss.cisco.com.domain: 9217+
PTR? 187.0.1.2.in-addr.arpa. (43)
19:20:45.681090 < gss.cisco.com.domain > gssm.cisco.com.49165: 9217
NXDomain* 0/1/0 (111)
...
```

Specifying Name Servers

The GSS can communicate with a maximum of eight name servers for name and address resolution. Use the **ip name server** command to specify the IP address of one or more name servers, to a maximum of eight name servers. To disable one or more name servers, use the **no** form of this command.

The syntax for this command is as follows:

ip name-server *ip-addresses*

The ip-addresses variable identifies the IP addresses for the name servers. You can enter a maximum of eight name servers, separated by spaces. Enter each IP address in dotted-decimal notation.

To configure the IP address of a single name server, enter:

```
gssm1.cisco.com(config)# ip name-server 172.16.17.18
```

To configure the IP addresses of multiple name servers, enter:

```
gssm1.cisco.com(config)# ip name-server 172.16.17.18 192.168.2.22
172.16.1.2
```

The GSS requires a functioning nameserver to operate properly and perform DNS resolutions. If the nameserver is not properly configured using the **ip name-server** command, or if the configured nameservers are not reachable for any reason (down, network loss, or a firewall), the GSS will not be able to perform DNS resolutions when you attempt to log in. In this case, the timeout may take several minutes. This behavior occurs when you attempt to log in through a Telnet, SSH, or FTP connection.

To enable the GSS to perform DNS resolution, always configure more than one nameserver. For example:

```
gss.example.com(config)#ip name-server 192.168.1.1
gss.example.com(config)#ip name-server 192.168.2.2
gss.example.com(config)#ip name-server 192.168.3.3
```

This behavior may also occur if you configure access lists for the GSS. In this case, create access lists that allow the DNS responses from a nameserver. For example:

```
gss.example.com(config)#access-list ac11 permit udp any eq 53
```

Another solution is to limit incoming DNS response packets only from your configured nameservers (more secure). For example:

```
gss.example.com(config)#access-list ac11 permit udp 192.168.1.1
255.255.255.255 eq 53
gss.example.com(config)#access-list ac12 permit udp 192.168.1.2
255.255.255.255 eq 53
gss.example.com(config)#access-list ac13 permit udp 192.168.1.3
255.255.255.255 eq 53
```

Configuring an IP Route for the GSS

To establish IP connectivity to the GSS, configure a static IP route to connect the GSS to next hop router using the **ip route** command. A static route consists of a destination network address and mask and the next hop IP address to reach the destination. You can also specify a default route using the **ip default-gateway** command to direct IP packets for which no other destination is listed in the routing table. The default gateway is useful for forwarding otherwise unrouteable packets by the GSS.

Use the following **ip** command options to configure a static or default IP route:

- **ip default-gateway**—Defines a default gateway. The GSS uses the default gateway to route IP packets when there is no specific route found to the destination. To delete the IP default gateway, use the **no** form of this command.
- **ip route**— Adds a specific static route for a network host. Any IP packet designated for the specified host uses the configured route. To disable an IP routing, use the **no** form of this command.

The syntax for the **ip** command is:

```
ip { default-gateway ip-address | route destination_address netmask
      gateway }
```

```
no ip { default-gateway ip-address | route destination_address netmask
        gateway }
```

The keywords and arguments are as follows:

- **default-gateway**—Specifies the default gateway to route IP packets when there is no specific route found to the destination.
- *ip-address*—Specifies an IP address for the default gateway. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- **route**—Specifies the network route.
- *destination_address*—Specifies the destination IP route address. Enter the IP address in dotted-decimal notation.
- *netmask*—Specifies the subnet mask. Enter the subnet mask in dotted-decimal notation (for example, 255.255.255.0).
- *gateway*—Specifies the gateway IP address. Enter the IP address in dotted-decimal notation.

For example, to configure a default gateway, enter:

```
gssml.cisco.com(config)# ip default-gateway 192.168.7.18
```

For example, to configure a static IP network route, enter:

```
gssml.cisco.com(config)# ip route 172.16.227.128 172.16.227.250
```

To display the IP routing table for the GSS, use the **show ip routes** command.

```
gssml.cisco.com# show ip routes
```

Table 4-4 describes the fields in the **show ip routes** output.

Table 4-4 Field Descriptions for show ip routes Command

Field	Description
User Defined Routes	Static IP routes configured for the GSS.
Kernel IP Routing Table	IP routing information for the GSS.

Table 4-4 Field Descriptions for *show ip routes* Command (continued)

Field	Description
Destination	Destination network or destination host.
Gateway	Gateway address (or 0.0.0.0 if no gateway address is set).
Genmask	Subnet mask for the destination network.
Flags	Possible flags are as follows: <ul style="list-style-type: none"> • U (route is up) • H (target is a host) • G (use gateway) • R (reinstate route for dynamic routing) • D (dynamically installed by daemon or redirect) • M (modified from routing daemon or redirect) • A (installed by addrconf) • C (cache entry) • ! (reject route)
Metric	Distance to the target, usually counted in hops.
Ref	Number of references to this route.
Use	Count of lookups for the route.
Iface	Interface to which packets for this route will be sent.

Resolving a Host or Domain Name to an IP Address

To resolve a host or domain name to an IP address, use the **dnslookup** command. The syntax for this command is as follows:

```
dnslookup {hostname | domainname}
```

The variables are as follows:

- *hostname*—Name of the host on the network.

- *domainname*—Name of the domain.

In the example, the **dnslookup** command resolves the hostname **myhost.cisco.com** to IP address 172.16.69.11.

```
gssm1.cisco.com# dnslookup myhost.cisco.com
Server: mydnsserver.cisco.com
Address: 172.16.69.12

Name: myhost.cisco.com
Address: 172.16.69.11
```

Configuring a Primary GSSM

The primary GSSM performs content routing as well as centralized management functions for the GSS network. The primary GSSM serves as the organizing point of the GSS network, hosting the embedded GSS database that contains configuration information for all of your GSS resources, such as individual GSS devices and DNS rules. Other GSS devices report their status to the primary GSSM. The primary GSSM offers a single, centralized GUI for monitoring and administering your entire GSS network.

A typical GSS deployment may contain a maximum of eight GSS devices on a corporate intranet or the Internet. At least one GSS—and no more than two GSS devices—must be configured as the primary GSSM and standby GSSM. The primary GSSM monitors the other GSS devices on the network and offers features for managing and monitoring request routing services using a GUI accessible through secure HTTP. Only one primary GSSM can be “active” at any time, with the second GSSM serving as a “standby,” or backup device.

Before you configure request routing or add GSS devices to your GSS network, first configure and enable a primary GSSM. After you have configured a primary GSSM, you may optionally configure a different GSS as the standby (redundant) GSSM.

Use the **gss enable gssm-primary** command to create the embedded database on the primary GSSM. This command also performs the other initialization processes to enable the device in a network of GSS devices. Enabling a GSS device is a one-time initialization step that is required only when you first set up the device within a network of GSS devices.

To configure a GSS device as a primary GSSM, perform the following steps:

1. Log in to the CLI of the GSS device and enable privileged EXEC mode. GSS configuration requires that you enter into privileged EXEC mode on the CLI. Ensure that your login has adequate permissions.

```
gssm1.example.com> enable
gssm1.example.com#
```

2. Enter the **gss enable gssm-primary** command to configure your GSS device as the primary GSSM in the GSS network.

```
gssm1.example.com# gss enable gssm-primary
```



Note When you use the **gss enable gssm-primary** command and a database exists on this GSS device, an error message appears. If this error message appears, use the **gss disable** command to remove the existing configuration and return the GSS device to its initial state, which includes deleting the GSSM database from the GSS device.

3. Save your configuration changes to memory.

```
gssm1.example.com# copy running-config startup-config
```

If you fail to save your configuration changes, the GSS device reverts to its previous settings upon a reboot.

At this point, you can access the GUI on the primary GSSM. After logging in to the primary GSSM GUI, use it to activate the standby GSSM and GSS devices on your network, as described in [Chapter 5, Activating GSS Devices from the GUI](#).

Configuring a Standby GSSM

The standby GSSM performs GSLB functions for the GSS network even while operating in standby mode. In addition, the standby GSSM can be configured to act as the GSSM should the primary GSSM need to go offline for repair or maintenance, or becomes unavailable to communicate with other GSS devices. As with the primary GSSM, the standby GSSM is configured to run the GSSM GUI and contains a duplicate copy of the embedded GSS database that is currently installed on the primary GSSM. Any configuration or network changes affecting the GSS network are synchronized between the primary and the standby GSSM.

The switching of roles between the designated primary GSSM and the standby GSSM is intended to be a temporary GSS network configuration until the original primary GSSM is back online. For details about changing the GSSM role in your GSS network, see the *Cisco Global Site Selector Administration Guide*, Chapter 1, Managing GSS Devices from the GUI.

To configure a GSS device as a standby GSSM, perform the following steps:

1. If you have not done so already, configure and enable a primary GSSM as described in the “Configuring a Primary GSSM” section.
2. Log in to the CLI of the GSS device and enable privileged EXEC mode. GSS configuration requires that you enter into privileged EXEC mode on the CLI. Ensure that your login has adequate permissions to do so.

```
gssm2.example.com> enable
gssm2.example.com#
```

3. Enter the **gss enable gssm-standby** command to enable your standby GSSM device and direct it to the primary GSSM in your GSS network. This command registers the standby GSSM with the primary GSSM.

The syntax for this command is as follows:

```
gss enable gssm-standby primary_GSSM_hostname |
primary_GSSM_IP_address
```

The variables are:

- *primary_GSSM_hostname*—DNS hostname of the device currently serving as the primary GSSM
- *primary_GSSM_IP_address*—DNS hostname of the device currently serving as the primary GSSM

For example, to enable gss2.example.com as the standby GSSM and direct it to the primary GSSM, gssm1.example.com, enter:

```
gssm2.example.com# gss enable gssm-standby gssm1.example.com
```

4. Save your configuration changes to memory.

```
gssm1.example.com# copy running-config startup-config
```

If you fail to save your configuration changes, the GSS device reverts to its previous settings upon a reboot.

Configuring a Global Site Selector

The GSS performs routing of DNS queries based on DNS rules and conditions configured using the primary GSSM. Each GSS is known to and synchronized with the GSSM, but individual GSS devices do not report their presence or status to the other. Each GSS on your network delegates authority to the GSS devices that serve DNS requests. Each GSS is managed separately using the Cisco CLI. GUI support is not available on a GSS device.

To configure a GSS device, perform the following steps:

1. Log in to the CLI of the GSS device and enable privileged EXEC mode.

```
gss1.example.com> enable
gss1.example.com#
```



Note GSS configuration requires that you enter into privileged EXEC mode on the CLI. Ensure that your login has adequate permissions to do so.

2. Use the **gss enable** command to enable your GSS device as a GSS and direct it to the primary GSSM in your GSS network. Specify either the domain name or the network address of the primary GSSM.

The syntax for this command is as follows:

```
gss enable gss primary_GSSM_hostname | primary_GSSM_IP_address
```

The variables are:

- *primary_GSSM_hostname*—The DNS hostname of the device currently serving as the primary GSSM
- *primary_GSSM_IP_address*—The DNS hostname of the device currently serving as the primary GSSM

For example, to enable gss.example.com as a GSS and direct it to the primary GSSM, gssm1.example.com, enter:

```
gss.example.com# gss enable gss gssm1.example.com
```

3. Save your configuration changes to memory.

```
gss1.example.com# copy running-config startup-config
```

If you fail to save your configuration changes, the device reverts to its previous settings upon a reboot.

Where to Go Next

To activate and register your standby GSSM and GSS devices from the primary GSSM GUI, proceed to [Chapter 5, Activating GSS Devices from the GUI](#). This chapter also describes how to log in to the primary GSSM GUI.

You can also activate and configure your GSS devices from the CLI instead. To do so, proceed to [Chapter 6, Activating GSS Devices from the CLI](#).