



Assigning the Switch IP Address and Default Gateway

This chapter describes how to create the initial switch configuration (for example, assigning the IP address and default gateway information) by using a variety of automatic and manual methods. It also describes how to modify the switch startup configuration. Unless otherwise noted, the term *switch* refers to a standalone switch and to a switch stack.



Note

For complete syntax and usage information for the commands used in this chapter, see the command reference for this release and the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*.

This chapter consists of these sections:

- [Understanding the Boot Process, page 3-1](#)
- [Assigning Switch Information, page 3-2](#)
- [Checking and Saving the Running Configuration, page 3-10](#)
- [Modifying the Startup Configuration, page 3-12](#)
- [Scheduling a Reload of the Software Image, page 3-16](#)



Note

Information in this chapter about configuring IP addresses and DHCP is specific to IP Version 4 (IPv4). If you plan to enable IP Version 6 (IPv6) forwarding on your switch, see [Chapter 36, “Configuring IPv6 Unicast Routing”](#) for information specific to IPv6 address format and configuration. To enable IPv6, the stack switch must be running the advanced IP services image.

Understanding the Boot Process

To start your switch, you need to follow the procedures in the *Getting Started Guide* or the hardware installation guide for installing and powering on the switch and setting up the initial switch configuration (IP address, subnet mask, default gateway, secret and Telnet passwords, and so forth).

The normal boot process involves the operation of the boot loader software, which performs these activities:

- Performs low-level CPU initialization. It initializes the CPU registers, which control where physical memory is mapped, its quantity, its speed, and so forth.
- Performs power-on self-test (POST) for the CPU subsystem. It tests the CPU DRAM and the portion of the flash device that makes up the flash file system.
- Initializes the flash file system on the system board.
- Loads a default operating system software image into memory and boots the switch.

The boot loader provides access to the flash file system before the operating system is loaded. Normally, the boot loader is used only to load, uncompress, and launch the operating system. After the boot loader gives the operating system control of the CPU, the boot loader is not active until the next system reset or power-on.

The boot loader also provides trap-door access into the system if the operating system has problems serious enough that it cannot be used. The trap-door mechanism provides enough access to the system so that if it is necessary, you can format the flash file system, reinstall the operating system software image by using the Xmodem Protocol, recover from a lost or forgotten password, and finally restart the operating system. For more information, see the [“Recovering from a Software Failure”](#) section on page 43-2 and the [“Recovering from a Lost or Forgotten Password”](#) section on page 43-3.



Note

You can disable password recovery. For more information, see the [“Disabling Password Recovery”](#) section on page 9-5.

Before you can assign switch information, make sure you have connected a PC or terminal to the console port, and configured the PC or terminal-emulation software baud rate and character format to match these of the switch console port:

- Baud rate default is 9600.
- Data bits default is 8.



Note If the data bits option is set to 8, set the parity option to none.

- Stop bits default is 1.
- Parity settings default is none.

Assigning Switch Information

You can assign IP information through the switch setup program, through a DHCP server, or manually.

Use the switch setup program if you want to be prompted for specific IP information. With this program, you can also configure a hostname and an enable secret password. It gives you the option of assigning a Telnet password (to provide security during remote management) and configuring your switch as a command or member switch of a cluster or as a standalone switch. For more information about the setup program, see the hardware installation guide.

The switch stack is managed through a single IP address. The IP address is a system-level setting and is not specific to the stack master or to any other stack member. You can still manage the stack through the same IP address even if you remove the stack master or any other stack member from the stack, provided there is IP connectivity.

**Note**

Stack members retain their IP address when you remove them from a switch stack. To avoid a conflict by having two devices with the same IP address in your network, change the IP address of the switch that you removed from the switch stack.

Use a DHCP server for centralized control and automatic assignment of IP information after the server is configured.

**Note**

If you are using DHCP, do not respond to any of the questions in the setup program until the switch receives the dynamically assigned IP address and reads the configuration file.

If you are an experienced user familiar with the switch configuration steps, manually configure the switch. Otherwise, use the setup program described previously.

These sections contain this configuration information:

- [Default Switch Information, page 3-3](#)
- [Understanding DHCP-Based Autoconfiguration, page 3-3](#)
- [Manually Assigning IP Information, page 3-10](#)

Default Switch Information

Table 3-1 shows the default switch information.

Table 3-1 *Default Switch Information*

Feature	Default Setting
IP address and subnet mask	No IP address or subnet mask are defined.
Default gateway	No default gateway is defined.
Enable secret password	No password is defined.
Hostname	The factory-assigned default hostname is <i>Switch</i> .
Telnet password	No password is defined.
Cluster command switch functionality	Disabled.
Cluster name	No cluster name is defined.

Understanding DHCP-Based Autoconfiguration

DHCP provides configuration information to Internet hosts and internetworking devices. This protocol consists of two components: one for delivering configuration parameters from a DHCP server to a device and a mechanism for allocating network addresses to devices. DHCP is built on a client-server model, in which designated DHCP servers allocate network addresses and deliver configuration parameters to dynamically configured devices. The switch can act as both a DHCP client and a DHCP server.

During DHCP-based autoconfiguration, your switch (DHCP client) is automatically configured at startup with IP address information and a configuration file.

With DHCP-based autoconfiguration, no DHCP client-side configuration is needed on your switch. However, you need to configure the DHCP server for various lease options associated with IP addresses. If you are using DHCP to relay the configuration file location on the network, you might also need to configure a Trivial File Transfer Protocol (TFTP) server and a Domain Name System (DNS) server.

**Note**

We recommend a redundant connection between a switch stack and the DHCP, DNS, and TFTP servers. This is to help ensure that these servers remain accessible in case one of the connected stack members is removed from the switch stack.

The DHCP server for your switch can be on the same LAN or on a different LAN than the switch. If the DHCP server is running on a different LAN, you should configure a DHCP relay device between your switch and the DHCP server. A relay device forwards broadcast traffic between two directly connected LANs. A router does not forward broadcast packets, but it forwards packets based on the destination IP address in the received packet.

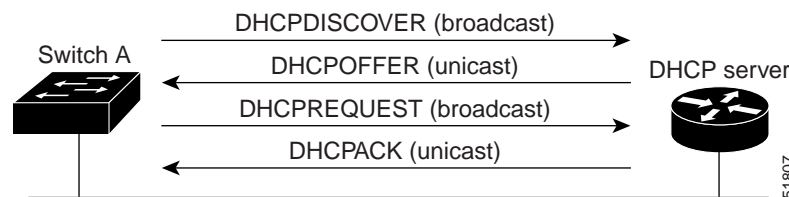
DHCP-based autoconfiguration replaces the BOOTP client functionality on your switch.

DHCP Client Request Process

When you boot your switch, the DHCP client is invoked and requests configuration information from a DHCP server when the configuration file is not present on the switch. If the configuration file is present and the configuration includes the **ip address dhcp** interface configuration command on specific routed interfaces, the DHCP client is invoked and requests the IP address information for those interfaces.

Figure 3-1 shows the sequence of messages that are exchanged between the DHCP client and the DHCP server.

Figure 3-1 DHCP Client and Server Message Exchange



The client, Switch A, broadcasts a DHCPDISCOVER message to locate a DHCP server. The DHCP server offers configuration parameters (such as an IP address, subnet mask, gateway IP address, DNS IP address, a lease for the IP address, and so forth) to the client in a DHCPOFFER unicast message.

In a DHCPREQUEST broadcast message, the client returns a formal request for the offered configuration information to the DHCP server. The formal request is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client. With this message, the client and server are bound, and the client uses configuration information received from the server. The amount of information the switch receives depends on how you configure the DHCP server. For more information, see the “[Configuring the TFTP Server](#)” section on page 3-6.

If the configuration parameters sent to the client in the DHCPOFFER unicast message are invalid (a configuration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server sends the client a DHCPNAK denial broadcast message, which means that the offered configuration parameters have not been assigned, that an error has occurred during the negotiation of the parameters, or that the client has been slow in responding to the DHCPOFFER message (the DHCP server assigned the parameters to another client).

A DHCP client might receive offers from multiple DHCP or BOOTP servers and can accept any of the offers; however, the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address is allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address. If the switch accepts replies from a BOOTP server and configures itself, the switch broadcasts, instead of unicasts, TFTP requests to obtain the switch configuration file.

Configuring DHCP-Based Autoconfiguration

These sections contain this configuration information:

- [DHCP Server Configuration Guidelines, page 3-5](#)
- [Configuring the TFTP Server, page 3-6](#)
- [Configuring the DNS, page 3-6](#)
- [Configuring the Relay Device, page 3-7](#)
- [Obtaining Configuration Files, page 3-7](#)
- [Example Configuration, page 3-8](#)

If your DHCP server is a Cisco device, see the “Configuring DHCP” section of the “IP Addressing and Services” section of the *Cisco IOS IP Configuration Guide, Release 12.2* for additional information about configuring DHCP.

DHCP Server Configuration Guidelines

Follow these guidelines if you are configuring a device as a DHCP server:

You should configure the DHCP server with reserved leases that are bound to each switch by the switch hardware address.

If you want the switch to receive IP address information, you must configure the DHCP server with these lease options:

- IP address of the client (required)
- Subnet mask of the client (required)
- DNS server IP address (optional)
- Router IP address (default gateway address to be used by the switch) (required)

If you want the switch to receive the configuration file from a TFTP server, you must configure the DHCP server with these lease options:

- TFTP server name (required)
- Boot filename (the name of the configuration file that the client needs) (recommended)
- Hostname (optional)

Depending on the settings of the DHCP server, the switch can receive IP address information, the configuration file, or both.

If you do not configure the DHCP server with the lease options described previously, it replies to client requests with only those parameters that are configured. If the IP address and the subnet mask are not in the reply, the switch is not configured. If the router IP address or the TFTP server name are not found, the switch might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not affect autoconfiguration.

Configuring the TFTP Server

Based on the DHCP server configuration, the switch attempts to download one or more configuration files from the TFTP server. If you configured the DHCP server to respond to the switch with all the options required for IP connectivity to the TFTP server, and if you configured the DHCP server with a TFTP server name, address, and configuration filename, the switch attempts to download the specified configuration file from the specified TFTP server.

If you did not specify the configuration filename, the TFTP server, or if the configuration file could not be downloaded, the switch attempts to download a configuration file by using various combinations of filenames and TFTP server addresses. The files include the specified configuration filename (if any) and these files: `network-config`, `cisconet.cfg`, `hostname.config`, or `hostname.cfg`, where *hostname* is the switch's current hostname. The TFTP server addresses used include the specified TFTP server address (if any) and the broadcast address (255.255.255.255).

For the switch to successfully download a configuration file, the TFTP server must contain one or more configuration files in its base directory. The files can include these files:

- The configuration file named in the DHCP reply (the actual switch configuration file).
- The `network-config` or the `cisconet.cfg` file (known as the default configuration files).
- The `router-config` or the `ciscotr.cfg` file (These files contain commands common to all switches. Normally, if the DHCP and TFTP servers are properly configured, these files are not accessed.)

If you specify the TFTP server name in the DHCP server-lease database, you must also configure the TFTP server name-to-IP-address mapping in the DNS-server database.

If the TFTP server to be used is on a different LAN from the switch, or if it is to be accessed by the switch through the broadcast address (which occurs if the DHCP server response does not contain all the required information described previously), a relay must be configured to forward the TFTP packets to the TFTP server. For more information, see the [“Configuring the Relay Device” section on page 3-7](#). The preferred solution is to configure the DHCP server with all the required information.

Configuring the DNS

The DHCP server uses the DNS server to resolve the TFTP server name to an IP address. You must configure the TFTP server name-to-IP address map on the DNS server. The TFTP server contains the configuration files for the switch.

You can configure the IP addresses of the DNS servers in the lease database of the DHCP server from where the DHCP replies will retrieve them. You can enter up to two DNS server IP addresses in the lease database.

The DNS server can be on the same or on a different LAN as the switch. If it is on a different LAN, the switch must be able to access it through a router.

Configuring the Relay Device

You must configure a relay device, also referred to as a *relay agent*, when a switch sends broadcast packets that require a response from a host on a different LAN. Examples of broadcast packets that the switch might send are DHCP, DNS, and in some cases, TFTP packets. You must configure this relay device to forward received broadcast packets on an interface to the destination host.

If the relay device is a Cisco router, enable IP routing (**ip routing** global configuration command), and configure helper addresses by using the **ip helper-address** interface configuration command.

For example, in [Figure 3-2](#), configure the router interfaces as follows:

On interface 10.0.0.2:

```
router(config-if)# ip helper-address 20.0.0.2
router(config-if)# ip helper-address 20.0.0.3
router(config-if)# ip helper-address 20.0.0.4
```

On interface 20.0.0.1

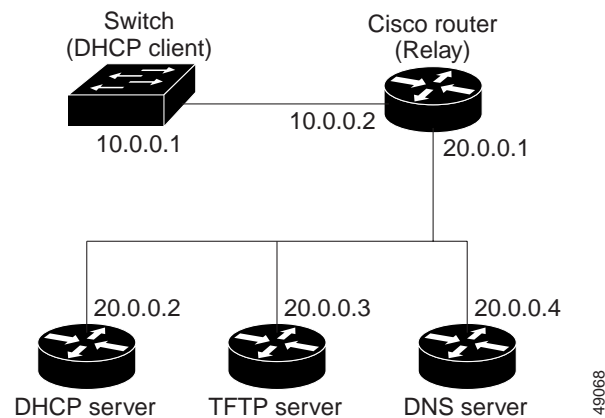
```
router(config-if)# ip helper-address 10.0.0.1
```



Note

If the switch is acting as the relay device, configure the interface as a routed port. For more information, see the [“Routed Ports” section on page 11-4](#) and the [“Configuring Layer 3 Interfaces” section on page 11-25](#).

Figure 3-2 Relay Device Used in Autoconfiguration



Obtaining Configuration Files

Depending on the availability of the IP address and the configuration filename in the DHCP reserved lease, the switch obtains its configuration information in these ways:

- The IP address and the configuration filename is reserved for the switch and provided in the DHCP reply (one-file read method).

The switch receives its IP address, subnet mask, TFTP server address, and the configuration filename from the DHCP server. The switch sends a unicast message to the TFTP server to retrieve the named configuration file from the base directory of the server and upon receipt, it completes its boot-up process.

- The IP address and the configuration filename is reserved for the switch, but the TFTP server address is not provided in the DHCP reply (one-file read method).

The switch receives its IP address, subnet mask, and the configuration filename from the DHCP server. The switch sends a broadcast message to a TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, it completes its boot-up process.

- Only the IP address is reserved for the switch and provided in the DHCP reply. The configuration filename is not provided (two-file read method).

The switch receives its IP address, subnet mask, and the TFTP server address from the DHCP server. The switch sends a unicast message to the TFTP server to retrieve the network-confg or cisco.net.cfg default configuration file. (If the network-confg file cannot be read, the switch reads the cisco.net.cfg file.)

The default configuration file contains the hostnames-to-IP-address mapping for the switch. The switch fills its host table with the information in the file and obtains its hostname. If the hostname is not found in the file, the switch uses the hostname in the DHCP reply. If the hostname is not specified in the DHCP reply, the switch uses the default *Switch* as its hostname.

After obtaining its hostname from the default configuration file or the DHCP reply, the switch reads the configuration file that has the same name as its hostname (*hostname-confg* or *hostname.cfg*, depending on whether network-confg or cisco.net.cfg was read earlier) from the TFTP server. If the cisco.net.cfg file is read, the filename of the host is truncated to eight characters.

If the switch cannot read the network-confg, cisco.net.cfg, or the hostname file, it reads the router-confg file. If the switch cannot read the router-confg file, it reads the ciscotr.cfg file.


Note

The switch broadcasts TFTP server requests if the TFTP server is not obtained from the DHCP replies, if all attempts to read the configuration file through unicast transmissions fail, or if the TFTP server name cannot be resolved to an IP address.

Example Configuration

Figure 3-3 shows a sample network for retrieving IP information by using DHCP-based autoconfiguration.

Figure 3-3 DHCP-Based Autoconfiguration Network Example

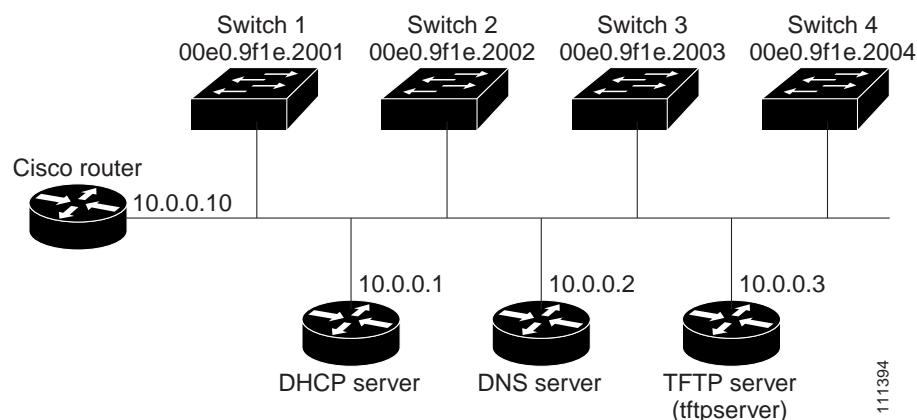


Table 3-2 shows the configuration of the reserved leases on the DHCP server.

Table 3-2 DHCP Server Configuration

	Switch A	Switch B	Switch C	Switch D
Binding key (hardware address)	00e0.9f1e.2001	00e0.9f1e.2002	00e0.9f1e.2003	00e0.9f1e.2004
IP address	10.0.0.21	10.0.0.22	10.0.0.23	10.0.0.24
Subnet mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Router address	10.0.0.10	10.0.0.10	10.0.0.10	10.0.0.10
DNS server address	10.0.0.2	10.0.0.2	10.0.0.2	10.0.0.2
TFTP server name	<i>tftpserver</i> or <i>10.0.0.3</i>	<i>tftpserver</i> or <i>10.0.0.3</i>	<i>tftpserver</i> or <i>10.0.0.3</i>	<i>tftpserver</i> or <i>10.0.0.3</i>
Boot filename (configuration file) (optional)	switcha-confg	switchb-confg	switchc-confg	switchd-confg
Hostname (optional)	switcha	switchb	switchc	switchd

DNS Server Configuration

The DNS server maps the TFTP server name *tftpserver* to IP address 10.0.0.3.

TFTP Server Configuration (on UNIX)

The TFTP server base directory is set to */tftpserver/work/*. This directory contains the *network-confg* file used in the two-file read method. This file contains the hostname to be assigned to the switch based on its IP address. The base directory also contains a configuration file for each switch (*switcha-confg*, *switchb-confg*, and so forth) as shown in this display:

```
prompt> cd /tftpserver/work/
prompt> ls
network-confg
switcha-confg
switchb-confg
switchc-confg
switchd-confg
prompt> cat network-confg
ip host switcha 10.0.0.21
ip host switchb 10.0.0.22
ip host switchc 10.0.0.23
ip host switchd 10.0.0.24
```

DHCP Client Configuration

No configuration file is present on Switch A through Switch D.

Configuration Explanation

In Figure 3-3, Switch A reads its configuration file as follows:

- It obtains its IP address 10.0.0.21 from the DHCP server.
- If no configuration filename is given in the DHCP server reply, Switch A reads the *network-confg* file from the base directory of the TFTP server.
- It adds the contents of the *network-confg* file to its host table.
- It reads its host table by indexing its IP address 10.0.0.21 to its hostname (switcha).
- It reads the configuration file that corresponds to its hostname; for example, it reads *switch1-confg* from the TFTP server.

Switches B through D retrieve their configuration files and IP addresses in the same way.

Manually Assigning IP Information

Beginning in privileged EXEC mode, follow these steps to manually assign IP information to multiple switched virtual interfaces (SVIs):



Note

If the switch is running the IP services image, you can also manually assign IP information to a port if you first put the port into Layer 3 mode by using the **no switchport** interface configuration command.

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface vlan <i>vlan-id</i>	Enter interface configuration mode, and enter the VLAN to which the IP information is assigned. The VLAN range is 1 to 4094.
Step 3	ip address <i>ip-address subnet-mask</i>	Enter the IP address and subnet mask.
Step 4	exit	Return to global configuration mode.
Step 5	ip default-gateway <i>ip-address</i>	Enter the IP address of the next-hop router interface that is directly connected to the switch where a default gateway is being configured. The default gateway receives IP packets with unresolved destination IP addresses from the switch. Once the default gateway is configured, the switch has connectivity to the remote networks with which a host needs to communicate. Note When your switch is configured to route with IP, it does not need to have a default gateway set.
Step 6	end	Return to privileged EXEC mode.
Step 7	show interfaces vlan <i>vlan-id</i>	Verify the configured IP address.
Step 8	show ip redirects	Verify the configured default gateway.
Step 9	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove the switch IP address, use the **no ip address** interface configuration command. If you are removing the address through a Telnet session, your connection to the switch will be lost. To remove the default gateway address, use the **no ip default-gateway** global configuration command.

For information on setting the switch system name, protecting access to privileged EXEC commands, and setting time and calendar services, see [Chapter 7, “Administering the Switch.”](#)

Checking and Saving the Running Configuration

You can check the configuration settings that you entered or changes that you made by entering this privileged EXEC command:

```
Switch# show running-config
Building configuration...
```

```
Current configuration: 1363 bytes
```

```

!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Stack1
!
enable secret 5 $1$ej9.$DMUvAUnZOAmvmgqBEzIxEO
!
.
<output truncated>
.
interface gigabitethernet6/0/1
  no switchport
  ip address 172.20.137.50 255.255.255.0
!
interface gigabitethernet6/0/2
  mvr type source

<output truncated>

...!
interface VLAN1
  ip address 172.20.137.50 255.255.255.0
  no ip directed-broadcast
!
ip default-gateway 172.20.137.1 !
!
snmp-server community private RW
snmp-server community public RO
snmp-server community private@es0 RW
snmp-server community public@es0 RO
snmp-server chassis-id 0x12
!
end

```

To store the configuration or changes you have made to your startup configuration in flash memory, enter this privileged EXEC command:

```

Switch# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...

```

This command saves the configuration settings that you made. If you fail to do this, your configuration will be lost the next time you reload the system. To display information stored in the NVRAM section of flash memory, use the **show startup-config** or **more startup-config** privileged EXEC command.

For more information about alternative locations from which to copy the configuration file, see [Appendix C, “Working with the Cisco IOS File System, Configuration Files, and Software Images.”](#)

Modifying the Startup Configuration

These sections describe how to modify the switch startup configuration:

- [Default Boot Configuration, page 3-12](#)
- [Automatically Downloading a Configuration File, page 3-12](#)
- [Booting Manually, page 3-13](#)
- [Booting a Specific Software Image, page 3-14](#)
- [Controlling Environment Variables, page 3-15](#)

See also “Switch Stack Configuration Files” section on page 5-14 and Appendix C, “Working with the Cisco IOS File System, Configuration Files, and Software Images,” for information about switch stack configuration files.

Default Boot Configuration

[Table 3-3](#) shows the default boot configuration.

Table 3-3 *Default Boot Configuration*

Feature	Default Setting
Operating system software image	<p>The switch attempts to automatically boot the system using information in the BOOT environment variable. If the variable is not set, the switch attempts to load and execute the first executable image it can by performing a recursive, depth-first search throughout the flash file system.</p> <p>The Cisco IOS image is stored in a directory that has the same name as the image file (excluding the .bin extension).</p> <p>In a depth-first search of a directory, each encountered subdirectory is completely searched before continuing the search in the original directory.</p>
Configuration file	<p>Configured switches use the <i>config.text</i> file stored on the system board in flash memory.</p> <p>A new switch has no configuration file.</p>

Automatically Downloading a Configuration File

You can automatically download a configuration file to your switch by using the DHCP-based autoconfiguration feature. For more information, see the “[Understanding DHCP-Based Autoconfiguration](#)” section on page 3-3.

Specifying the Filename to Read and Write the System Configuration

By default, the Cisco IOS software uses the file *config.text* to read and write a nonvolatile copy of the system configuration. However, you can specify a different filename, which will be loaded during the next boot cycle.



Note This command only works properly from a standalone switch.

Beginning in privileged EXEC mode, follow these steps to specify a different configuration filename:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	boot config-file flash:/file-url	Specify the configuration file to load during the next boot cycle. For <i>file-url</i> , specify the path (directory) and the configuration filename. Filenames and directory names are case sensitive.
Step 3	end	Return to privileged EXEC mode.
Step 4	show boot	Verify your entries. The boot config-file global configuration command changes the setting of the CONFIG_FILE environment variable.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no boot config-file** global configuration command.

Booting Manually

By default, the switch automatically boots; however, you can configure it to manually boot.



Note This command only works properly from a standalone switch.

Beginning in privileged EXEC mode, follow these steps to configure the switch to manually boot during the next boot cycle:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	boot manual	Enable the switch to manually boot during the next boot cycle.
Step 3	end	Return to privileged EXEC mode.

	Command	Purpose
Step 4	show boot	<p>Verify your entries.</p> <p>The boot manual global command changes the setting of the MANUAL_BOOT environment variable.</p> <p>The next time you reboot the system, the switch is in boot loader mode, shown by the <i>switch:</i> prompt. To boot the system, use the boot filesystem:/file-url boot loader command.</p> <ul style="list-style-type: none"> For <i>filesystem:</i>, use flash: for the system board flash device. For <i>file-url</i>, specify the path (directory) and the name of the bootable image. <p>Filenames and directory names are case sensitive.</p>
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To disable manual booting, use the **no boot manual** global configuration command.

Booting a Specific Software Image

By default, the switch attempts to automatically boot the system using information in the BOOT environment variable. If this variable is not set, the switch attempts to load and execute the first executable image it can by performing a recursive, depth-first search throughout the flash file system. In a depth-first search of a directory, each encountered subdirectory is completely searched before continuing the search in the original directory. However, you can specify a specific image to boot.

Beginning in privileged EXEC mode, follow these steps to configure the switch to boot a specific image during the next boot cycle:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	boot system filesystem:/file-url	<p>Configure the switch to boot a specific image in flash memory during the next boot cycle.</p> <ul style="list-style-type: none"> For <i>filesystem:</i>, use flash: for the system board flash device. For <i>file-url</i>, specify the path (directory) and the name of the bootable image. <p>If you enter this command on a stack master, the specified software image is loaded only on the stack master during the next boot cycle.</p> <p>Filenames and directory names are case sensitive.</p>
Step 3	boot system switch { <i>number</i> all }	<p>(Optional) Specify the switch members on which the system image is loaded during the next boot cycle:</p> <ul style="list-style-type: none"> Use <i>number</i> to specify a stack member. Use all to specify all stack members.
Step 4	end	Return to privileged EXEC mode.

	Command	Purpose
Step 5	show boot	Verify your entries. The boot system global command changes the setting of the BOOT environment variable. During the next boot cycle, the switch attempts to automatically boot the system using information in the BOOT environment variable.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no boot system** global configuration command.

Controlling Environment Variables

With a normally operating switch, you enter the boot loader mode only through a switch console connection configured for 9600 bps. Unplug the switch power cord, and press the switch **Mode** button while reconnecting the power cord. You can release the **Mode** button a second or two after the LED above port 1 turns off. Then the boot loader *switch:* prompt appears.

The switch boot loader software provides support for nonvolatile environment variables, which can be used to control how the boot loader, or any other software running on the system, behaves. Boot loader environment variables are similar to environment variables that can be set on UNIX or DOS systems.

Environment variables that have values are stored in flash memory outside of the flash file system.

Each line in these files contains an environment variable name and an equal sign followed by the value of the variable. A variable has no value if it is not listed in this file; it has a value if it is listed in the file even if the value is a null string. A variable that is set to a null string (for example, “”) is a variable with a value. Many environment variables are predefined and have default values.

Environment variables store two kinds of data:

- Data that controls code, which does not read the Cisco IOS configuration file. For example, the name of a boot loader helper file, which extends or patches the functionality of the boot loader can be stored as an environment variable.
- Data that controls code, which is responsible for reading the Cisco IOS configuration file. For example, the name of the Cisco IOS configuration file can be stored as an environment variable.

You can change the settings of the environment variables by accessing the boot loader or by using Cisco IOS commands. Under normal circumstances, it is not necessary to alter the setting of the environment variables.



Note

For complete syntax and usage information for the boot loader commands and environment variables, see the command reference for this release.

Table 3-4 describes the function of the most common environment variables.

Table 3-4 Environment Variables

Variable	Boot Loader Command	Cisco IOS Global Configuration Command
BOOT	<p>set BOOT <i>filesystem:/file-url ...</i></p> <p>A semicolon-separated list of executable files to try to load and execute when automatically booting. If the BOOT environment variable is not set, the system attempts to load and execute the first executable image it can find by using a recursive, depth-first search through the flash file system. If the BOOT variable is set but the specified images cannot be loaded, the system attempts to boot the first bootable file that it can find in the flash file system.</p>	<p>boot system { <i>filesystem:/file-url ...</i> switch { <i>number</i> all } }</p> <p>Specifies the Cisco IOS image to load during the next boot cycle and the stack members on which the image is loaded. This command changes the setting of the BOOT environment variable.</p>
MANUAL_BOOT	<p>set MANUAL_BOOT yes</p> <p>Decides whether the switch automatically or manually boots.</p> <p>Valid values are 1, yes, 0, and no. If it is set to no or 0, the boot loader attempts to automatically boot the system. If it is set to anything else, you must manually boot the switch from the boot loader mode.</p>	<p>boot manual</p> <p>Enables manually booting the switch during the next boot cycle and changes the setting of the MANUAL_BOOT environment variable.</p> <p>The next time you reboot the system, the switch is in boot loader mode. To boot the system, use the boot flash:<i>filesystem:/file-url</i> boot loader command, and specify the name of the bootable image.</p>
CONFIG_FILE	<p>set CONFIG_FILE flash:<i>/file-url</i></p> <p>Changes the filename that Cisco IOS uses to read and write a nonvolatile copy of the system configuration.</p>	<p>boot config-file flash:<i>/file-url</i></p> <p>Specifies the filename that Cisco IOS uses to read and write a nonvolatile copy of the system configuration. This command changes the CONFIG_FILE environment variable.</p>
SWITCH_NUMBER	<p>set SWITCH_NUMBER <i>stack-member-number</i></p> <p>Changes the member number of a stack member.</p>	<p>switch <i>current-stack-member-number</i> renumber <i>new-stack-member-number</i></p> <p>Changes the member number of a stack member.</p>
SWITCH_PRIORITY	<p>set SWITCH_PRIORITY <i>stack-member-number</i></p> <p>Changes the priority value of a stack member.</p>	<p>switch <i>stack-member-number</i> priority <i>priority-number</i></p> <p>Changes the priority value of a stack member.</p>

Scheduling a Reload of the Software Image

You can schedule a reload of the software image to occur on the switch at a later time (for example, late at night or during the weekend when the switch is used less), or you can synchronize a reload network-wide (for example, to perform a software upgrade on all switches in the network).



Note

A scheduled reload must take place within approximately 24 days.

Configuring a Scheduled Reload

To configure your switch to reload the software image at a later time, use one of these commands in privileged EXEC mode:

- **reload in** *[hh:]mm [text]*

This command schedules a reload of the software to take affect in the specified minutes or hours and minutes. The reload must take place within approximately 24 days. You can specify the reason for the reload in a string up to 255 characters in length.

To reload a specific switch in a switch stack, use the **reload slot** *stack-member-number* privileged EXEC command.

- **reload at** *hh:mm [month day | day month] [text]*

This command schedules a reload of the software to take place at the specified time (using a 24-hour clock). If you specify the month and day, the reload is scheduled to take place at the specified time and date. If you do not specify the month and day, the reload takes place at the specified time on the current day (if the specified time is later than the current time) or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight.



Note Use the **at** keyword only if the switch system clock has been set (through Network Time Protocol (NTP), the hardware calendar, or manually). The time is relative to the configured time zone on the switch. To schedule reloads across several switches to occur simultaneously, the time on each switch must be synchronized with NTP.

The **reload** command halts the system. If the system is not set to manually boot, it reboots itself. Use the **reload** command after you save the switch configuration information to the startup configuration (**copy running-config startup-config**).

If your switch is configured for manual booting, do not reload it from a virtual terminal. This restriction prevents the switch from entering the boot loader mode and thereby taking it from the remote user's control.

If you modify your configuration file, the switch prompts you to save the configuration before reloading. During the save operation, the system requests whether you want to proceed with the save if the CONFIG_FILE environment variable points to a startup configuration file that no longer exists. If you proceed in this situation, the system enters setup mode upon reload.

This example shows how to reload the software on the switch on the current day at 7:30 p.m:

```
Switch# reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 1996 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

This example shows how to reload the software on the switch at a future time:

```
Switch# reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 1996 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

To cancel a previously scheduled reload, use the **reload cancel** privileged EXEC command.

Displaying Scheduled Reload Information

To display information about a previously scheduled reload or to find out if a reload has been scheduled on the switch, use the **show reload** privileged EXEC command.

It displays reload information including the time the reload is scheduled to occur and the reason for the reload (if it was specified when the reload was scheduled).