CHAPTER 3

Configuring the Switch for the First Time

This chapter describes how to initially configure the Catalyst 4006 switch with Supervisor Engine III and supplements the administration information and procedures in these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/index.htm
- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.1, at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/index.htm

This chapter consists of the following sections:

Note For complete syntax and usage information for the commands used in this publication, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

Default Switch Configuration

Table 3-1 shows the default switch configuration.

Table 3-1 Default Switch Configuration

Feature	Default Value
Administrative connection	Normal mode
Global switch information	No value for the following: • System name • System contact • Location
System clock	No value for system clock time

Table 3-1 Default Switch Configuration (continued)

Feature	Default Value
Passwords	No passwords are configured for normal mode or enable mode (press the **Return** key)
Switch prompt	`Switch>`
Interfaces	Enabled, with speed and flow control auto-negotiated, and without IP addresses

Configuring the Switch

These sections describe how to configure the switch:

Using Configuration Mode

You can configure your switch from configuration mode as follows:

Step 1 Connect a console terminal to the console interface of your supervisor engine.

Step 2 After a few seconds, you will see the user EXEC prompt (`Switch>`). Type **enable** to enter enable mode:

```
Switch> enable
```

Note Configuration changes can be made only in enable mode.

The prompt will change to the privileged EXEC prompt (#):

```
Switch#
```

Step 3 At the prompt (#), enter the **configure terminal** command to enter configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

At the prompt, enter the **interface** *type slot/interface* command to enter interface configuration mode:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)#
```

In either of these configuration modes, you can enter changes to the switch configuration. Enter the **end** command to exit configuration mode.

Step 4 Save your settings. (See the "Saving the Running Configuration Settings" section on page 3-3.)

Your switch is now minimally configured and can boot-up with the configuration you entered. To see a list of the configuration commands, enter **?** at the prompt or press the **help** key in configuration mode.

Checking the Running Configuration Before Saving

You can verify the configuration settings that you entered or changes that you made by entering the **show running-config** command at the privileged EXEC prompt (#), as shown in the following example:

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

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch

<...output truncated...>

!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end
Switch#
```

Saving the Running Configuration Settings

To store the configuration, or changes to the configuration or to the startup configuration in NVRAM, enter the **copy running-config startup-config** command at the privileged EXEC prompt (#) as follows:

```
Switch# copy running-config startup-config
```

This command saves the configuration settings that you created in configuration mode. If you fail to do this step, your configuration will be lost the next time you reload the system.

Reviewing the Configuration

To display information stored in NVRAM, enter the **show startup-config** EXEC command.

The following sample output shows a typical system configuration:

```
Switch# show startup-config
Using 1579 out of 491500 bytes, uncompressed size = 7372 bytes
Uncompressed configuration from 1579 bytes to 7372 bytes
```

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
!
ip subnet-zero
!
!
!
!
interface GigabitEthernet1/1
 no snmp trap link-status
!
interface GigabitEthernet1/2
 no snmp trap link-status
!--More--

<...output truncated...>

!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

Configuring a Default Gateway

Note The switch uses the default gateway only when it is not configured with a routing protocol.

To send data to subnets other than its own when the switch is not configured with a routing protocol, configure a default gateway. The default gateway must be the IP address of an interface on a router that is directly connected to the switch.

To configure a default gateway, perform this task:

	Task	Command
Step 1	Configure a default gateway.	`Switch(config)# ip default-gateway A.B.C.D`
Step 2	Verify that the default gateway displays correctly in the IP routing table.	`Switch# show ip route`

This example shows how to configure a default gateway and how to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip default-gateway 172.20.52.35
Switch(config)# end
3d17h: %SYS-5-CONFIG_I: Configured from console by console
Switch# show ip route
Default gateway is 172.20.52.35

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
Switch#
```

Configuring a Static Route

If your Telnet station or SNMP network management workstation is on a different network from your switch and a routing protocol has not been configured, you might need to add a static routing table entry for the network where your end station is located.

To configure a static route, perform this task:

	Task	Command
Step 1	Configure a static route to the remote network.	`Switch(config)# ip route dest_IP_address mask {forwarding_IP | vlan vlan_ID}`
Step 2	Verify that the static route displays correctly.	`Switch# show running-config`

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.10.5.10 on the switch with a subnet mask and IP address 172.20.3.35 of the forwarding router:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip route 171.10.5.10 255.255.255.255 172.20.3.35
Switch(config)# end
Switch#
```

This example shows how to use the **show running-config** command to confirm the configuration of the static route:

```
Switch# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.10.5.10 255.255.255.255 172.20.3.35
no ip http server
!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
```

```
end

Switch#
```

This example shows how to use the **ip route** command to configure the static route IP address 171.20.5.3 with subnet mask and connected over VLAN 1 to a workstation on the switch:

```
Switch# configure terminal
Switch(config)# ip route 171.20.5.3 255.255.255.255 vlan 1
Switch(config)# end
Switch#
```

This example shows how to use the **show running-config** command to confirm the configuration of the static route:

```
Switch# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.20.52.3 255.255.255.255 Vlan1
no ip http server
!
!
x25 host z
!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

Protecting Access to Privileged EXEC Commands

The tasks in following sections provide a way to use a password to control access to the system configuration file and privileged EXEC commands:

Setting or Changing a Static Enable Password

To set or change a static password that controls access to the privileged EXEC mode, enter this command:

Command	Purpose
`Switch(config)# enable password password`	Sets a new password or change an existing password for the privileged EXEC mode.

This example shows how to configure an enable password as "lab" at the privileged EXEC mode:

```
Switch# configure terminal
Switch(config)# enable password lab
Switch(config)#
```

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Using the enable password and enable secret Commands

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands. Both commands configure an encrypted password that you must enter to access the enable mode (the default) or any other privilege level that you specify.

We recommend that you use the **enable secret** command.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure the switch to require an enable password, perform either of these tasks:

	Task	Command
Step 1	Establish a password for the privileged EXEC mode.	`Switch(config)# enable password [level level] {password | encryption-type encrypted-password}`
Step 2	Specify a secret password, saved using a nonreversible encryption method. (If **enable password** and **enable secret** commands are both set, users must enter the enable secret password.)	`Switch(config)# enable secret [level level] {password | encryption-type encrypted-password}`

When you enter either of these password commands with the **level** option, you define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** configuration command to specify commands accessible at various levels.

If you enable the **service password-encryption** command, the password you enter is encrypted. When you display it with the **more system:running-config** command, it displays the password in encrypted form.

If you specify an encryption type, you must provide an encrypted password—an encrypted password you copy from another Catalyst 4006 switch with Supervisor Engine III configuration.

Note You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the "Recovering a Lost Enable Password" section on page 3-11 if you lose or forget your password.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Setting or Changing a Privileged Password

To set or change a privileged password, enter this command:

Command	Purpose
`Switch(config-line)# password password`	Sets a new password or change an existing password for the privileged level.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Setting TACACS+ Password Protection for Privileged EXEC Mode

For complete information about TACACS+ and Radius, refer to these publications:

- "Authentication, Authorization, and Accounting (AAA)," chapter in *Cisco IOS Security Configuration Guide*, Release 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt1/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

To set the TACACS+ protocol to determine whether a user can access privileged EXEC mode, enter this command:

Command	Purpose
`Switch(config)# enable use-tacacs`	Sets the TACACS-style user ID and password-checking mechanism for the privileged EXEC mode.

When you set TACACS password protection at the privileged EXEC mode, the **enable** EXEC command prompts for a new username and a new password. This information is then passed to the TACACS+ server for authentication. If you are using the extended TACACS+, it also passes any existing UNIX user identification code to the TACACS+ server.

Note When used without extended TACACS, the **enable use-tacacs** command allows anyone with a valid username and password to access the privileged EXEC mode, creating a potential security risk. This problem occurs because the query resulting from entering the **enable** command is indistinguishable from an attempt to log in without extended TACACS+.

Encrypting Passwords

Because protocol analyzers can examine packets (and read passwords), you can increase access security by configuring the IOS software to encrypt passwords. Encryption prevents the password from being readable in the configuration file.

To configure the IOS software to encrypt passwords, enter this command:

Command	Purpose
`Switch(config)# service password-encryption`	Encrypts a password.

Encryption occurs when the current configuration is written or when a password is configured. Password encryption is applied to all passwords, including authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol (BGP) neighbor passwords. The **service password-encryption** command keeps unauthorized individuals from viewing your password in your configuration file.

Caution The **service password-encryption** command does not provide a high level of network security. If you use this command, you should also take additional network security measures.

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can regain control of the switch after having lost or forgotten the encrypted password. See the "Recovering a Lost Enable Password" section on page 3-11 if you lose or forget your password.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Configuring Multiple Privilege Levels

By default, the IOS software has two modes of password security: user EXEC mode and privileged EXEC mode. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password fairly widely. If you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to more restricted users.

The tasks in the following sections describe how to configure additional levels of security:

Setting the Privilege Level for a Command

To set the privilege level for a command, perform this task:

	Task	Command
Step 1	Set the privilege level for a command.	`Switch(config)# privilege mode level level command`
Step 2	Specify the enable password for a privilege level.	`Switch(config)# enable password level level [encryption-type] password`

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or a group of lines, enter this command:

Command	Purpose
`Switch(config-line)# privilege level level`	Changes the default privilege level for the line.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-10.

Logging In to a Privilege Level

To log in at a specified privilege level, enter this command:

Command	Purpose
`Switch# enable level`	Logs in to a specified privilege level.

Exiting a Privilege Level

To exit to a specified privilege level, enter this command:

Command	Purpose
`Switch# disable level`	Exits to a specified privilege level.

Displaying the Password, Access Level, and Privilege Level Configuration

To display detailed password information, perform this task:

	Task	Command
Step 1	Display the password and access level, and privilege level configuration.	`Switch# show running-config`
Step 2	Show the privilege level configuration.	`Switch# show privilege`

This example shows how to display the password and access level configuration:

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

Current configuration:
!
version 12.0
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname Switch
!
boot system flash sup-bootflash
enable password lab
!
<...output truncated...>
```

This example shows how to display the privilege level configuration:

```
Switch# show privilege
Current privilege level is 15
Switch#
```

Recovering a Lost Enable Password

Perform these steps to recover a lost enable password:

Step 1 Connect to the console interface.

Step 2 Stop the boot sequence and enter ROM monitor by pressing **Ctrl-C** during the first 5 seconds of boot-up.

Step 3 Configure the switch to boot-up without reading the configuration memory (NVRAM). See Configuring the Software Configuration Register, page 3-12 for more information.

Step 4 Reboot the system.

Step 5 Access enable mode (this can be done without a password if a password has not been configured).

Step 6 View or change the password, or erase the configuration.

Step 7 Reconfigure the switch to boot-up and read the NVRAM as it normally does.

Step 8 Reboot the system.

Modifying the Supervisor Engine Startup Configuration

These sections describe how the startup configuration on the supervisor engine works and how to modify the configuration register and BOOT variable:

- BOOTLDR Environment Variable, page 3-18

Understanding the Supervisor Engine Boot Configuration

The supervisor engine boot-up process involves two software images: ROM monitor and supervisor engine software. When the switch is booted or reset, the ROM-monitor code is executed. Depending on the NVRAM configuration, the supervisor engine either stays in ROM-monitor mode or loads the supervisor engine software.

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the "Modifying the Boot Field and Using the boot Command" section on page 3-14. The BOOT environment variable is described in the "Specifying the Startup System Image" section on page 3-16.

Understanding the ROM Monitor

The ROM monitor is invoked upon switch boot-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROM-monitor mode. From ROM-monitor mode, you can manually load a software image from bootflash or a Flash Disk, or you can boot-up from the management interface. ROM-monitor mode loads a primary image, from which you can configure a secondary image to boot up from a specified source either locally or through the network using the BOOTLDR environment variable described in the "BOOTLDR Environment Variable" section on page 3-18.

You can also enter ROM-monitor mode by restarting the switch and then pressing **Ctrl-C** during the first 5 seconds of startup. If you are connected through a terminal server, you can escape to the Telnet prompt and enter the **send break** command to enter ROM-monitor mode.

Note **Ctrl-C** is always enabled for 5 seconds after rebooting the switch, regardless of whether the configuration-register setting has **Ctrl-C** disabled.

The ROM monitor has these features:

- Power-on confidence test
- Hardware initialization
- Boot capability (manual boot-upboot-up and autoboot)
- File system (read only while in ROMMON)

Configuring the Software Configuration Register

The switch uses a 16-bit software configuration register, which allows you to set specific system parameters. Settings for the software configuration register are written into NVRAM.

Following are some reasons for changing the software configuration register settings:

- To select a boot source and default boot filename
- To control broadcast addresses.
- To set the console terminal baud rate.

- To load operating software from Flash memory.
- To recover a lost password.
- To allow you to manually boot-up the system using the **boot** command at the bootstrap program prompt.
- To force an automatic boot-up from the system bootstrap software (boot image) or from a default system image in onboard Flash memory, and read any **boot system** commands that are stored in the configuration file in NVRAM.

Table 3-2 lists the meaning of each of the software configuration memory bits, and Table 3-3 defines the *boot field*.

Caution To avoid confusion and possibly halting the Catalyst 4006 switch with Supervisor Engine III, remember that valid configuration register settings might be combinations of settings and not just the individual settings listed in Table 3-2. For example, the factory default value of 0x0102 is a combination of settings.

Table 3-2 Software Configuration Register Bit Meaning

Bit Number[1]	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-3)
04	0x0010	Unused
05	0x0020	Bit two of Console line speed
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM[2] bit enabled
08	0x0100	Unused
09	0x0200	Unused
10	0x0400	Internet Protocol (IP) broadcast with all zeros
11 to 12	0x0800 to 0x1000	Bits one and zero of Console line speed (default is 9600 baud)
13	0x2000	Load ROM monitor after netboot fails
14	0x4000	IP broadcasts do not have network numbers

1. The factory default value for the configuration register is 0x0102. This value is a combination of the following: binary bit 8 = 0x0100 and binary bits 00 through 03 = 0x0002 (see Table 3-3).
2. OEM = original equipment manufacturer.

Table 3-3 Explanation of Boot Field (Configuration Register Bits 00 to 03)

Boot Field	Meaning
00	Stays at the system bootstrap prompt (does not autoboot)
01	Boots the first system image in onboard Flash memory
02 to 0F	Autoboots using image(s) specified by the BOOT environment variable.If more than one image is specified, the images are used in the order in which they appear in the boot variable. When the end of the list is reached, the first image is used again and the cycle repeats.

Modifying the Boot Field and Using the boot Command

The configuration register boot field determines whether the switch loads an operating system image, and if so, where it obtains this system image. The following sections describe using and setting the configuration register boot field, and the tasks you must perform to modify the configuration register boot field. At ROM Monitor, you can use the **confreg** command to modify the configuration register and change boot settings.

Bits 0 through 3 of the software configuration register contain the boot field.

Note The factory default configuration register setting for systems and spares is 0x0102.

When the boot field is set to either 0 or 1 (0-0-0-0 or 0-0-0-1), the system ignores any boot instructions in the system configuration file and the following occurs:

- When the boot field is set to 0, you must boot-up the operating system manually by issuing the **boot** command to the system bootstrap program or ROM monitor.
- When the boot field is set to 1, the system boots the first image in the bootflash single in-line memory module (SIMM).
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the switch loads the system image specified by **boot system** commands in the startup configuration file.

You can enter the **boot** command only, or enter the command and include additional boot instructions, such as the name of a file stored in Flash memory, or a file that you specify for booting from a network server. If you use the **boot** command without specifying a file or any other boot instructions, the system boots from the default Flash image (the first image in onboard Flash memory). Otherwise, you can instruct the system to boot up from a specific Flash image (using the **boot system flash** *filename* command).

You can also use the **boot** command to boot up images stored in the Compact Flash cards located in slot 0 on the supervisor engine.

Modifying the Boot Field

You modify the boot field from the software configuration register. To modify the software configuration register boot field, perform this task:

	Task	Command
Step 1	Determine the current configuration register setting.	`Switch# show version`
Step 2	Enter configuration mode, selecting the terminal option.	`Switch# configure terminal`
Step 3	Modify the existing configuration register setting to reflect the way in which you want the switch to load a system image.	`Switch(config)# config-register` *value*
Step 4	Exit configuration mode.	`Switch(config)# end`
Step 5	Reboot the switch to make your changes take effect.	`Switch# reload`

Perform the following procedure to modify the configuration register while the switch is running IOS:

Step 1 Enter the **enable** command and your password to enter privileged level, as follows:

```
Switch> enable
Password:
Switch#
```

Step 2 Enter the **configure terminal** command at the EXEC mode prompt (#), as follows:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

Step 3 Configure the configuration register to 0x102 as follows:

```
Switch(config)# config-register 0x102
```

Set the contents of the configuration register by entering the **config-register** *value* configuration command, where *value* is a hexadecimal number preceded by 0x (see Table 3-2 on page 3-13).

Step 4 Enter the **end** command to exit configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system software is reloaded by rebooting the system.

Step 5 Enter the **show version** EXEC command to display the configuration register value currently in effect, which will be used at the next reload. The value is displayed on the last line of the screen display, as shown in this example:

```
Configuration register is 0x141 (will be 0x102 at next reload)
```

Step 6 Save your settings. (See the "Saving the Running Configuration Settings" section on page 3-3. However, note that configuration register changes take effect only after the system reloads, such as when you enter a **reload** command from the console.)

Step 7 Reboot the system. The new configuration register value takes effect with the next system boot-up.

This completes the procedure for making configuration register changes.

Verifying the Configuration Register Setting

Enter the **show version** EXEC command to verify the current configuration register setting. In ROM-monitor mode, enter the **show version** command to verify the value of the configuration register boot field.

To verify the configuration register setting for the switch, enter this command:

Command	Purpose
`Switch# show version`	Displays the configuration register setting.

In this example, the **show version** command indicates that the current configuration register is set so that the switch does not automatically load an operating system image. Instead, it enters ROM-monitor mode and waits for you to enter ROM monitor commands.

```
Switch#show version
Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-IS-M), Experimental
Version 12.1(20010828:211314) [cisco 105]
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Thu 06-Sep-01 15:40 by
Image text-base:0x00000000, data-base:0x00ADF444
```

```
ROM:1.15
Switch uptime is 10 minutes
System returned to ROM by reload
Running default software

cisco Catalyst 4000 (MPC8240) processor (revision 3) with 262144K bytes
of memory.
Processor board ID Ask SN 12345
Last reset from Reload
Bridging software.
49 FastEthernet/IEEE 802.3 interface(s)
20 Gigabit Ethernet/IEEE 802.3 interface(s)
271K bytes of non-volatile configuration memory.

Configuration register is 0xEC60

Switch#
```

Specifying the Startup System Image

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image.

The BOOT environment variable is also described in the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images and Microcode" chapter of the IOS *Configuration Fundamentals Configuration Guide*.

Use the following sections to configure your switch to boot from Flash memory. Flash memory can be either be single in-line memory modules (SIMMs) or Flash Disks. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory.

Using Flash Memory

Flash memory allows you to do the following:

- Copy the system image to Flash memory using TFTP.
- Boot the system from Flash memory either automatically or manually.
- Copy the Flash memory image to a network server using TFTP or RCP.

Flash Memory Features

Flash memory features include the following:

- Can be remotely loaded with multiple system software images through TFTP or rcp transfers (one transfer for each file loaded).
- Allows you to boot a switch manually or automatically from a system software image stored in Flash memory. You can also boot directly from ROM.

Security Precautions

Note the following security precaution when loading from Flash memory:

Caution The system image stored in Flash memory can be changed only from privileged EXEC level on the console terminal.

Configuring Flash Memory

To configure your switch to boot from Flash memory, perform this procedure. Refer to the appropriate hardware installation and maintenance publication for complete instructions on installing the hardware.

Step 1 Copy a system image to Flash memory using TFTP or other protocols (refer to the "Cisco IOS File Management," "Loading and Maintaining System Images," chapters in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt2/fcd203.htm

Step 2 Configure the system to boot automatically from the desired file in Flash memory. You might need to change the configuration register value. See the "Modifying the Boot Field and Using the boot Command" section on page 3-14, for more information on modifying the configuration register.

Step 3 Save your configurations.

Step 4 Power cycle and reboot your system to ensure that all is working as expected.

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain commands. To create or modify the BOOT and BOOTLDR variables, use the **boot system** and **boot bootldr** global configuration commands, respectively.

Refer to the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images and Microcode" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the BOOT environment variable.

Note When you use the **boot system** and **boot bootldr** global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT and BOOTLDR variables using the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration and in the running configuration if a running configuration setting differs from a startup configuration setting.

This example shows how to check the BOOT and BOOTLDR variables on the switch:

```
Switch# show bootvar
BOOTLDR variable = bootflash:cat4000-is-mz,1;
Configuration register is 0x0

Switch#
```

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and file name that contains the boot loader image required to load system software. It defines the primary Cisco IOS image that will load the final image from another source.

Setting the BOOTLDR Environment Variable

To set the BOOTLDR environment variable, perform this task:

	Task	Command
Step 1	Verify that bootflash contains the boot loader image.	`Switch# dir bootflash:`
Step 2	Enter the configuration mode from the terminal.	`Switch# configure terminal`
Step 3	Set the BOOTLDR environment variable to specify the Flash device and file name of the boot loader image. This step modifies the runtime BOOTLDR environment variable.	`Switch(config)# boot bootldr bootflash:`*boot_loader*
Step 4	Exit configuration mode.	`Switch# end`
Step 5	Save this runtime BOOTLDR environment variable to your startup configuration.	`Switch# copy system:running-config nvram:startup-config`
Step 6	(Optional) Verify the contents of the BOOTLDR environment variable.	`Switch# show bootvar`

This example shows how to set the BOOTLDR variable:

```
Switch# dir bootflash:
Directory of bootflash:/

  1  -rw-     1599488   Nov 29 1999 11:12:29 cat4000-is-mz.XE.bin

15990784 bytes total (14391168 bytes free)
Switch# configure terminal
Switch (config)# boot bootldr bootflash:cat4000-is-mz.XE.bin
Switch (config)# end
Switch# copy system:running-config nvram:startup-config
[ok]
Switch# show bootvar
BOOTLDR variable = bootflash:cat4000-is-mz,1
Configuration register is 0x0
```