



Modifying the Switch Boot Configuration

This chapter describes how to modify the switch boot configuration, including the BOOT environment variable and the configuration register on the Catalyst enterprise LAN switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference—Catalyst 4000 Family, Catalyst 2948G, and Catalyst 2980G Switches*.

This chapter consists of these sections:

- [Understanding How the Switch Boot Configuration Works, page 28-1](#)
- [Default Switch Boot Configuration, page 28-4](#)
- [Setting the Configuration Register, page 28-5](#)
- [Setting the BOOT Environment Variable, page 28-7](#)
- [Setting and Clearing the CONFIG_FILE Environment Variable, page 28-8](#)
- [Displaying the Switch Boot Configuration, page 28-9](#)

Understanding How the Switch Boot Configuration Works

These sections describe how the boot configuration works on the Catalyst 4000 family, 2948G, and 2980G switches:

- [Understanding the Boot Process, page 28-2](#)
- [Understanding the ROM Monitor, page 28-2](#)
- [Understanding the Configuration Register, page 28-2](#)
- [Understanding the BOOT Environment Variable, page 28-3](#)
- [Understanding the CONFIG_FILE Environment Variable, page 28-4](#)

Understanding the Boot Process

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

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the “[Understanding the Configuration Register](#)” section on page 28-2. The BOOT environment variable is described in the “[Understanding the BOOT Environment Variable](#)” section on page 28-3.

Understanding the ROM Monitor

The ROM monitor code executes upon switch power-up, reset, or when a fatal exception occurs. The system enters ROM-monitor mode if the switch does not find a valid system 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 system image from Flash memory, from a network server file, or from bootflash.

**Note**

For complete syntax and usage information for the ROM monitor commands, refer to the *Command Reference—Catalyst 4000 Family, Catalyst 2948G, and Catalyst 2980G Switches*.

You can enter ROM-monitor mode by restarting the switch and then pressing **Ctrl-C** during the first five seconds of startup.

The following functionality is built into the ROM monitor:

- Power-on confidence test
- Hardware initialization
- Boot capability (allows manual boot and autoboot)
- Debug utility and crash analysis
- File system (the ROM monitor knows the simple file system and supports the newly developed file system through the dynamic linked file system library [MONLIB])
- Exception handling

Understanding the Configuration Register

The configuration register determines whether the switch loads an operating system image and where the system image is stored. The configuration register boot field determines if and how the ROM monitor loads a supervisor engine system image at startup. You can modify the boot field to force the switch to boot a particular system image at startup instead of using the default system image.

The lowest four bits (bits 3, 2, 1, and 0) of the 16-bit configuration register form the boot field. The default boot field value is 0x10F. The possible configuration register boot field settings are as follows:

- When the boot field equals 0000, the switch does not load a system image. The switch enters ROM-monitor mode from which you can enter ROM-monitor commands to manually load a system image.
- When the boot field equals 0001, the switch loads the first valid system image found in onboard Flash memory.
- When the boot field equals a value between 0010 and 1111, the switch loads the system image specified by **boot system** commands in the NVRAM configuration. It attempts to boot the image in the order in which you entered the **boot system** commands. If it cannot boot any image in the BOOT environment variable list, the switch remains in ROM-monitor mode. The exact booting sequence is defined by the ROM monitor.

The other bits in the configuration register function as follows when set:

- Bit 5 (0x0020): Enables CONFIG_FILE recurrence.
- Bit 6 (0x0040): Causes system software to clear NVRAM contents.
- Bit 7 (0x0080): Enables OEM bit (not used).
- Bit 8 (0x0100): Disables break.
- Bit 9 (0x0200): Uses secondary bootstrap (not used by the ROM monitor).
- Bit 10 (0x0400): Provides IP broadcast with all zeros (not used).
- Bits 11/12 (0x0800/0x1000): These bits are always set to 0/0 (9600 baud).
- Bit 13 (0x2000): Boots default Flash software if network boot fails (not used).
- Bit 14 (0x4000): IP broadcasts do not have network numbers (not used).
- Bit 15 (0x8000): Enables diagnostic messages and ignores NVRAM contents (not used).

Understanding the BOOT Environment Variable

The BOOT environment variable specifies a list of image files on various devices from which the switch can boot at startup.

You can add several images to the BOOT environment variable to provide a fail-safe boot configuration. If the first file fails to boot the switch, subsequent images specified in the BOOT variable are tried until the switch boots or there are no additional images to attempt to boot. If there is no valid image to boot, the system enters ROM-monitor mode where you can manually specify an image to boot.

The system stores and executes images in the order in which you added them to the BOOT variable. If you want to change the order in which images are tried at startup, you can either prepend and clear images from the BOOT variable to attain the desired order or you can clear the entire BOOT environment variable and then redefine the list in the desired order.

Understanding the CONFIG_FILE Environment Variable

In software release 5.2 and later, you can use the CONFIG_FILE environment variable to specify a list of configuration files on various devices to use to configure the switch at startup. You can specify one of the following functions:

- Nonrecurring—When you add a list of configuration files to the CONFIG_FILE environment variable, the next time the switch is restarted, the system erases the configuration in NVRAM and uses the specified files to configure the switch. The CONFIG_FILE variable is cleared before the switch is configured. Nonrecurring is the default setting.
- Recurring—When you add a list of configuration files to the CONFIG_FILE environment variable, the list is stored indefinitely in NVRAM. Each time the switch is restarted, the system erases the configuration in NVRAM and configures the switch using the configuration files specified. The CONFIG_FILE variable is not cleared.



Note

Remember that the CONFIG_FILE variable can be altered or its recurrence properties changed by commands in the configuration files used to configure the switch at startup.

For information on specifying recurrence or nonrecurrence, see the [“Setting CONFIG_FILE Recurrence” section on page 28-6](#).

When the switch boots up, if any of the files specified in the CONFIG_FILE environment variable are valid configuration files, the configuration in NVRAM is erased and the system uses the specified configuration file to configure the switch. If multiple valid configuration files are specified, each configuration file is executed in the order in which it appears in the CONFIG_FILE environment variable.

If any specified file is not a valid configuration file, the entry is skipped and subsequent files are tried until there are no additional images specified. If no valid configuration file is specified, the system retains the last configuration stored in NVRAM.



Note

For more information about using configuration files, see [Chapter 31, “Working with Configuration Files.”](#)

Default Switch Boot Configuration

[Table 28-1](#) shows the default switch boot configuration.

Table 28-1 Default Switch Boot Configuration

Feature	Default Configuration
Configuration register value	0x10f
Boot method	System boots from the image specified in the BOOT environment variable
ROM monitor console port baud rate	9600 baud ¹
ignore-config parameter	Disabled
BOOT environment variable	Empty

Table 28-1 Default Switch Boot Configuration (continued)

Feature	Default Configuration
CONFIG_FILE environment variable	bootflash:switch.cfg
CONFIG_FILE recurrence configuration register parameter	Nonrecurring

1. The ROM monitor console port baud rate is always 9600 baud.

Setting the Configuration Register

These sections describe how to modify the configuration register:

- [Setting the Boot Field in the Configuration Register, page 28-5](#)
- [Setting CONFIG_FILE Recurrence, page 28-6](#)
- [Setting the Switch to Ignore the NVRAM Configuration, page 28-6](#)

Setting the Boot Field in the Configuration Register

You can determine the boot method the switch will use at the next startup by setting the boot field in the configuration register. This command affects only the configuration register bits that control the boot field and leaves the remaining bits unaltered.

The following boot methods are supported:

- ROM monitor—Use the **rommon** keyword to keep the switch in ROM-monitor mode at startup.
- Bootflash—Use the **bootflash** keyword to cause the switch to boot from the first image stored in the onboard Flash.
- System—Use the **system** keyword to boot from the image specified in the BOOT environment variable (the default).



Note

We recommend that you use only the **rommon** and **system** options to the **set boot config-register boot** command.

To set the configuration register boot field, perform this task in privileged mode:

Task	Command
Specify the boot field in the configuration register.	set boot config-register boot {rommon bootflash system} [mod_num]

This example shows how to force the switch to enter ROM-monitor mode at the next startup:

```

Console> (enable) set boot config-register boot rommon
Configuration register is 0x0
ignore-config: disabled
auto-config: non-recurring
console baud: 9600
boot: the ROM monitor
Console> (enable)

```

Setting CONFIG_FILE Recurrence

By default, when you set the CONFIG_FILE environment variable, the list of configuration files to use at startup is retained only until the next time the switch is restarted.

You can cause the system software to retain the CONFIG_FILE environment variable settings indefinitely so that each time the switch is restarted, the specified configuration files are used to configure the switch.

This command affects only the configuration register bit that controls whether the CONFIG_FILE environment variable settings are recurring or nonrecurring. The remaining configuration register bits are unaltered.



Caution

With the CONFIG_FILE environment variable set to **recurring**, the current configuration in NVRAM is erased each time the switch is restarted and the switch is configured using the specified configuration files. With the CONFIG_FILE environment variable set to **non-recurring**, the current configuration in NVRAM is erased at the next restart and the switch is configured using the specified configuration files. The NVRAM configuration is retained after subsequent restarts (unless you again set the CONFIG_FILE variable).

To set the switch to retain the current CONFIG_FILE environment variable indefinitely, perform this task in privileged mode:

Task	Command
Set the switch to retain the current CONFIG_FILE environment variable indefinitely.	set boot config-register auto-config { recurring non-recurring }

This example shows how to set the switch to retain the current CONFIG_FILE variable indefinitely:

```
Console> (enable) set boot config-register auto-config recurring
Configuration register is 0x1820
ignore-config: disabled
auto-config: recurring
console baud: 9600
boot: the ROM monitor
Console> (enable)
```

Setting the Switch to Ignore the NVRAM Configuration

You can cause the system software to ignore the configuration information stored in NVRAM the next time the switch is restarted. This command affects only the configuration register bits that control whether the switch ignores the NVRAM configuration and leaves the remaining bits unaltered. This command only affects the next system restart.



Caution

Enabling the **ignore-config** parameter is the same as entering the **clear config all** command; that is, it clears the entire configuration stored in NVRAM the next time the switch is restarted.

To set the switch to ignore the NVRAM configuration at the next startup, perform this task in privileged mode:

Task	Command
Set the switch to ignore the contents of NVRAM at startup.	set boot config-register ignore-config enable

This example shows how to set the switch to ignore the NVRAM configuration at the next startup:

```
Console> (enable) set boot config-register ignore-config enable
Configuration register is 0x1860
ignore-config: enabled
auto-config: recurring
console baud: 9600
boot: the ROM monitor
Console> (enable)
```

Setting the BOOT Environment Variable

These sections describe how to modify the BOOT environment variable:

- [Setting the BOOT Environment Variable, page 28-7](#)
- [Clearing the BOOT Environment Variable Settings, page 28-8](#)

Setting the BOOT Environment Variable

To set the BOOT environment variable, perform this task in privileged mode:

Task	Command
Specify a system image to add to the BOOT environment variable.	set boot system flash <i>device:[filename]</i> [prepend] [<i>mod_num</i>]

This example shows how to add system images to the BOOT environment variable:

```
Console> (enable) set boot system flash bootflash:cat4000.5-1-1.bin
BOOT variable = bootflash:cat4000.5-1-1.bin,1;
Console> (enable) set boot system flash bootflash:cat4000.4-5-2.bin
BOOT variable = bootflash:cat4000.5-1-1.bin,1;bootflash:cat4000.4-5-2.bin,1;
Console> (enable) set boot system flash bootflash:cat4000.6-1-1.bin prepend
BOOT variable = bootflash:cat4000.6-1-1.bin,1;bootflash:cat4000.5-1-1.bin,1;
bootflash:cat4000.4-5-2.bin,1;
Console> (enable)
```

Clearing the BOOT Environment Variable Settings

To clear entries from the BOOT environment variable, perform one of these tasks in privileged mode:

Task	Command
<ul style="list-style-type: none"> Clear a specific image from the BOOT environment variable. 	clear boot system flash <i>device:[filename]</i> <i>[mod_num]</i>
<ul style="list-style-type: none"> Clear the entire BOOT environment variable. 	clear boot system all <i>[mod_num]</i>

This example shows how to clear a specific entry from the BOOT environment variable:

```
Console> (enable) clear boot system flash bootflash:cat4000.5-1-1.bin
BOOT variable = bootflash:cat4000.5-2-1.bin,1;bootflash:cat4000.4-5-2.bin,1;
Console> (enable)
```

This example shows how to clear the entire BOOT environment variable:

```
Console> (enable) clear boot system all
BOOT variable =
Console> (enable)
```

Setting and Clearing the CONFIG_FILE Environment Variable

These sections describe how to set and clear the CONFIG_FILE environment variable:



Note

For more information about using configuration files, see [Chapter 31, “Working with Configuration Files.”](#)

Setting the Variable

You can specify multiple configuration files with the **set boot auto-config** command by separating them with a semicolon (;). You must specify both the device name and the filename for each configuration file.



Note

You cannot prepend or append configuration files to the CONFIG_FILE environment variable. Entering the **set boot auto-config** command erases any list of configuration files previously specified using the **set boot auto-config** command.

To set the CONFIG_FILE environment variable, perform this task in privileged mode (depending on your supervisor engine and switch type):

Task	Command
Specify the list of configuration files to add to the CONFIG_FILE environment variable.	set boot auto-config <i>device:filename[;device:filename...]</i>

This example shows how to add a list of configuration files to the CONFIG_FILE environment variable:

```
Console> (enable) set boot auto-config bootflash:generic.cfg;bootflash:4003_1_noc.cfg
CONFIG_FILE variable = bootflash:generic.cfg;bootflash:4003_1_noc.cfg
WARNING: nvram configuration may be lost during next bootup,
        and re-configured using the file(s) specified.
Console> (enable)
```

Clearing the Variable Settings

To clear the entries from the CONFIG_FILE environment variable, perform this task in privileged mode:

Task	Command
Clear the entries in the CONFIG_FILE environment variable.	clear boot auto-config

This example shows how to clear the entries in the CONFIG_FILE environment variable:

```
Console> (enable) clear boot auto-config
CONFIG_FILE variable =
Console> (enable)
```

Displaying the Switch Boot Configuration

To display the current configuration register, BOOT environment variable, and CONFIG_FILE environment variable settings, perform this task in privileged mode:

Task	Command
Display the current configuration register, BOOT environment variable, and CONFIG_FILE environment variable settings.	show boot [<i>mod_num</i>]

This example shows how to display the current configuration register, BOOT environment variable, and CONFIG_FILE environment variable settings:

```
Console> (enable) show boot
BOOT variable = bootflash:cat4000.5-2-1.bin,1;
CONFIG_FILE variable = bootflash:generic.cfg;bootflash:4003_1_noc.cfg

Configuration register is 0x12f
ignore-config: disabled
auto-config: recurring
console baud: 9600
boot: image specified by the boot system commands

Console> (enable)
```

