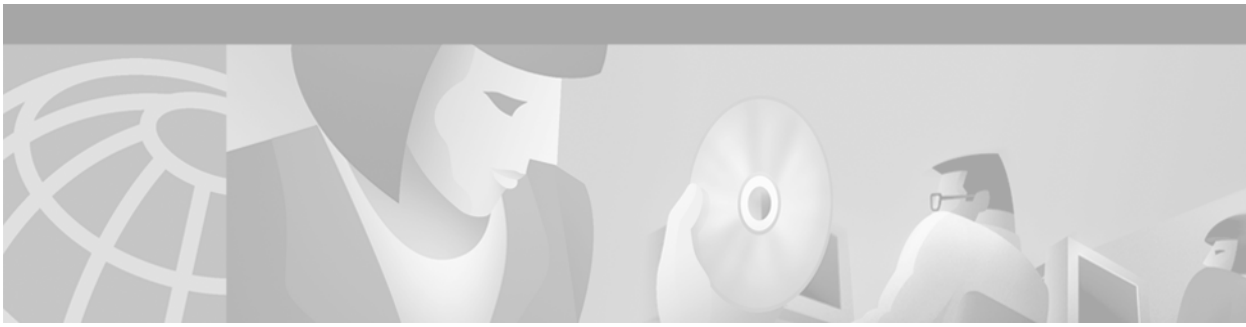


Rebooting a Router

This chapter describes the basic procedure a router follows when it reboots, how to alter the procedure, and how to use the ROM Monitor.

For a complete description of the booting commands mentioned in this chapter, refer to the "Booting Commands" chapter in the *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Rebooting a Router Task List

You can perform the tasks related to rebooting discussed in the following sections:

- Displaying Booting Information
- Rebooting Procedures
- Modifying the Configuration Register Boot Field
- Setting Environment Variables
- Scheduling a Reload of the System Image
- Entering ROM Monitor Mode
- Manually Loading a System Image from ROM Monitor
- Configuring High System Availability on the Cisco 7500 Series

Displaying Booting Information

Use the following commands in EXEC mode to display information about system software, system image files, and configuration files:

Command	Purpose
`show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`more nvram:startup-config`	Lists the startup configuration information. On all platforms except the Class A Flash file systems, the startup configuration is usually in NVRAM. On Class A Flash file systems, the CONFIG_FILE environment variable points to the startup configuration, defaulting to NVRAM.
`show version`	Lists the system software release version, system image name, configuration register setting, and other information.

Refer to the *Cisco IOS Configuration Fundamentals Command Reference* for examples of these commands.

You can also use the **o** command (the **confreg** command for some platforms) in ROM monitor mode to list the configuration register settings on some models.

Rebooting Procedures

The following sections describe what happens when the router reboots:

- What Configuration File Does the Router Use upon Startup?
- What Image Does the Router Use upon Startup?

What Configuration File Does the Router Use upon Startup?

On all platforms except Class A Flash file system platforms,

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software checks for configuration information in NVRAM.
  - If NVRAM holds valid configuration commands, the Cisco IOS software executes the commands automatically at startup.
  - If the software detects a problem with NVRAM or the configuration it contains (a CRC checksum error), it enters **setup** mode and prompts for configuration.

On Class A Flash file system platforms,

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software uses the configuration pointed to by the CONFIG_FILE environment variable.
  - When the CONFIG_FILE environment variable does not exist or is null (such as at first-time startup), the router uses NVRAM as the default startup device.
  - When the router uses NVRAM to start up and the system detects a problem with NVRAM or the configuration it contains, the router enters **setup** mode.

Problems can include a bad checksum for the information in NVRAM or an empty NVRAM with no configuration information. See the "Troubleshooting Hardware and Booting Problems" chapter publication *Internetwork Troubleshooting Guide* for troubleshooting procedures. See the "Using Setup for Configuration Changes" chapter in this publication for details on the **setup** command facility. For more information on environment variables, refer to the "Setting Environment Variables" section.

What Image Does the Router Use upon Startup?

When a router is powered on or rebooted, the following events happen:

- The ROM Monitor initializes.
- The ROM monitor checks the boot field (the lowest four bits) in the configuration register.
  - If the last digit of the boot field is 0 (for example, 0x100), the system does not boot an IOS image and waits for user intervention at the ROM Monitor prompt. From ROM monitor mode, you can manually boot the system using the **boot** or **b** command.
  - If the last digit of the boot field is1 (for example, 0x101), the boot helper image is loaded from ROM. (On some platforms, the boot helper image is specified by the BOOTLDR environment variable.)
  - If the last digit of the boot field is 2 through F (for example, 0x102 through 0x10F), the router boots the first valid image specified in the configuration file or specified by the BOOT environment variable.

Note The configuration register boot field value is expressed in hexidecimal. Since the boot field only encompasses the last four bits of the configuration register value, the only digit we are concerned with in this discussion is the last digit. The makes 0x1 (0000 0001) equivalent to 0x101 (1 0000 0001) in discussions of the boot field, as in both cases the last four bits are 0001.

When the boot field is 0x102 through 0x10F, the router goes through each **boot system** command in order until it boots a valid image. If bit 13 in the configuration register is set, each command will be tried once (bit 13 is indicated by the position occupied by *b* in the following hexidecimal notation: 0x*b*000). If bit 13 is not set, the **boot system** commands specifying a network server will be tried up to five more times. The timeouts between each consecutive attempt are 2, 4, 16, 256, and 300 seconds.

If the router cannot find a valid image, the following events happen:

- If all boot commands in the system configuration file specify booting from a network server and all commands fail, the system attempts to boot the first valid file in Flash memory.
- If the "boot-default-ROM-software" option in the configuration register is set, the router will start the boot image (the image contained in boot ROM or specified by the BOORLDR environment variable).
- If the "boot-default-ROM-software" option in the configuration register is not set, the system waits for user intervention at the ROM Monitor prompt. You must boot the router manually.
- If a fully functional system image is not found, the router will not function and must be reconfigured through a direct console port connection.

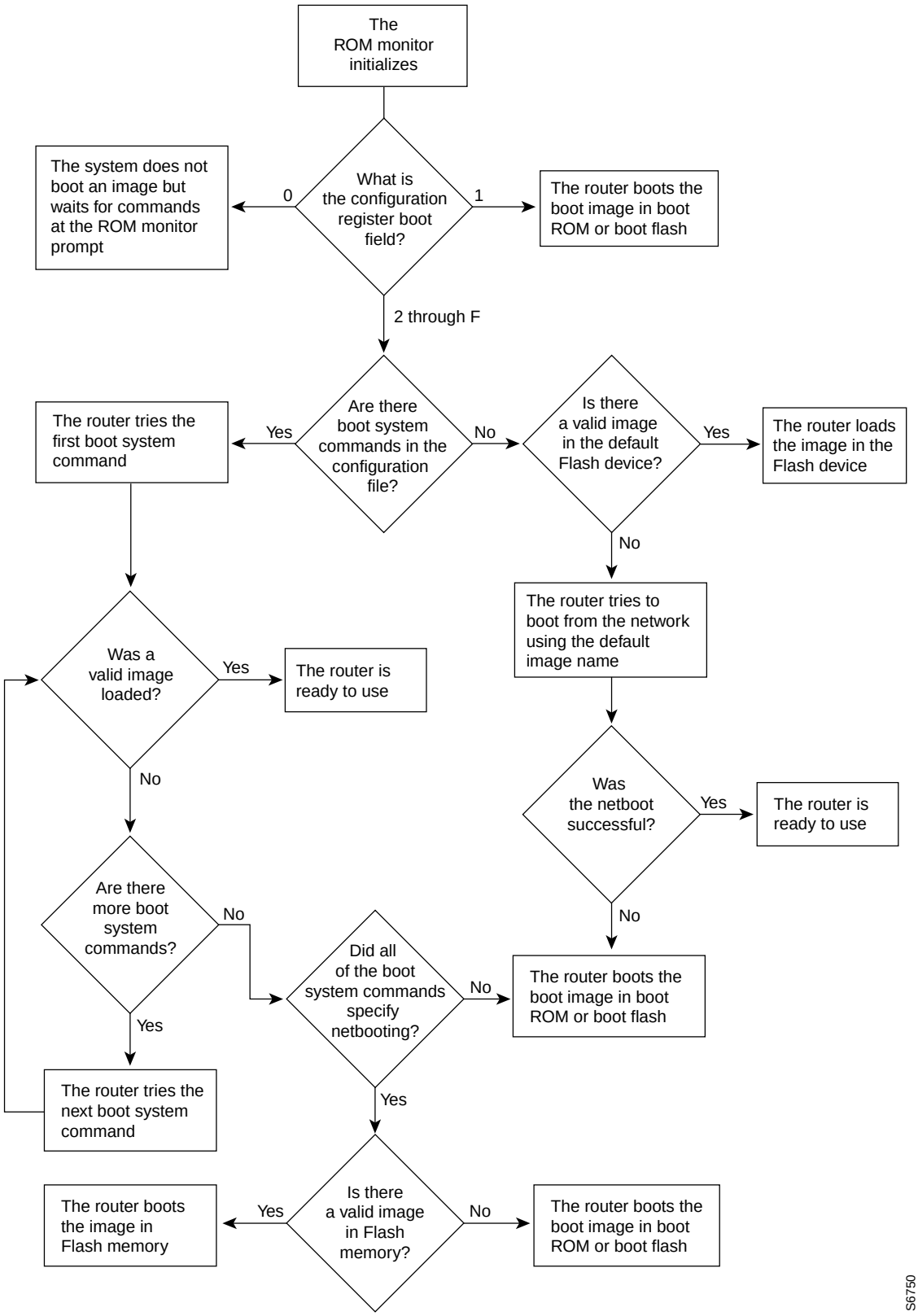
Note Refer to your platform documentation for information on the default location of the boot image.

When looking for a bootable file in Flash memory:

- The system searches for the filename in Flash memory. If a filename is not specified, the software searches through the entire Flash directory for a bootable file instead of picking only the first file.
- The system attempts to recognize the file in Flash memory. If the file is recognized, the software decides whether it is bootable by performing the following checks:
  - For run-from-Flash images, the software determines whether it is loaded at the correct execution address.
  - For run-from-RAM images, the software determines whether the system has enough RAM to execute the image.

Figure 11 illustrates the basic booting decision process.

Figure 11 Booting Process



Modifying the Configuration Register Boot Field

The configuration register boot field determines whether the router loads an operating system image, and if so, where it obtains this system image. This section contains the following topics:

- How the Router Uses the Boot Field
- Hardware Versus Software Configuration Register Boot Fields
- Modifying the Software Configuration Register Boot Field

Refer to the documentation for your platform for more information on the configuration register.

How the Router Uses the Boot Field

The lowest four bits of the 16-bit configuration register (bits 3, 2, 1, and 0) form the boot field. The following boot field values determine if the router loads an operating system and where it obtains the system image:

- When the entire boot field equals 0-0-0-0 (0x0), the router does not load a system image. Instead, it enters ROM monitor or "maintenance" mode from which you can enter ROM monitor commands to manually load a system image. Refer to the "Manually Loading a System Image from ROM Monitor" section for details on ROM monitor mode.
- When the entire boot field equals 0-0-0-1 (0x1), the router loads the boot helper or rxboot image.
- When the entire boot field equals a value between 0-0-1-0 (0x2) and 1-1-1-1 (0xF), the router loads the system image specified by **boot system** commands in the startup configuration file. When the startup configuration file does not contain **boot system** commands, the router tries to load a default system image stored on a network server.

  When loading a default system image from a network server, the router uses the configuration register settings to determine the default system image filename for booting from a network server. The router forms the default boot filename by starting with the word cisco and then appending the octal equivalent of the boot field number in the configuration register, followed by a hyphen (-) and the processor type name (cisco*nn-cpu*). See the appropriate hardware installation guide for details on the configuration register and the default filename.

Hardware Versus Software Configuration Register Boot Fields

You modify the boot field from either the hardware configuration register or the software configuration register, depending on the platform.

Most platforms have use a software configuration register. Refer to your hardware documentation for information on the configuration register for your platform.

The hardware configuration register can be changed only on the processor card with dual in-line package (DIP) switches located at the back of the router. For information on modifying the hardware configuration register, refer to the appropriate hardware installation guide.

Modifying the Software Configuration Register Boot Field

To modify the software configuration register boot field, use the following commands:

	Command	Purpose
Step 1	**show version**	Obtains the current configuration register setting. The configuration register is listed as a hexadecimal value.
Step 2	**configure terminal**	Enters configuration mode, selecting the terminal option.
Step 3	**config-register** *value*	Modifies the existing configuration register setting to reflect the way in which you want to load a system image. The configuration register value is in hexadecimal form with a leading "0x."
Step 4	**end**	Exits configuration mode.
Step 5	**show version**	Verifies that the configuration register setting is correct. Repeat steps 2 through 5 if the setting is not correct.
Step 6	**reload**	Reboots the router to make your changes take effect.

In ROM monitor mode, use the **o** command or the **confreg** command on some platforms to list the value of the configuration register boot field.

Modify the current configuration register setting to reflect the way in which you want to load a system image. To do so, change the least significant hexadecimal digit to one of the following:

- 0 to load the system image manually using the **boot** command in ROM monitor mode.
- 1 to load the system image from boot ROMs. On the Cisco 7200 series and Cisco 7500 series, this setting configures the system to automatically load the system image from bootflash.
- 2–F to load the system image from **boot system** commands in the startup configuration file or from a default system image stored on a network server.

For example, if the current configuration register setting is 0x101 and you want to load a system image from **boot system** commands in the startup configuration file, you would change the configuration register setting to 0x102.

Modifying the Software Configuration Register Boot Field Example

In the following example, the **show version** command indicates that the current configuration register is set so that the router does not automatically load an operating system image. Instead, it enters ROM monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the router to a load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version

Cisco Internetwork Operating System Software
IOS (tm) 4500 Software (C4500-J-M), Version 11.1(10.4), MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Mon 07-Apr-97 19:51 by dschwart
Image text-base: 0x600088A0, data-base: 0x60718000

ROM: System Bootstrap, Version 5.1(1) [daveu 1], RELEASE SOFTWARE (fc1)
FLASH: 4500-XBOOT Bootstrap Software, Version 10.1(1), RELEASE SOFTWARE (fc1)
```

```
Router1 uptime is 6 weeks, 5 days, 2 hours, 22 minutes
System restarted by error - a SegV exception, PC 0x6070F7AC
System image file is "c4500-j-mz.111-current", booted via flash

cisco 4500 (R4K) processor (revision 0x00) with 32768K/4096K bytes of memory.
Processor board ID 01242622
R4600 processor, Implementation 32, Revision 1.0
G.703/E1 software, Version 1.0.
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software (copyright 1994 by TGV Inc).
Basic Rate ISDN software, Version 1.0.
2 Ethernet/IEEE 802.3 interfaces.
2 Token Ring/IEEE 802.5 interfaces.
4 ISDN Basic Rate interfaces.
128K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
4096K bytes of processor board Boot flash (Read/Write)

Configuration register is 0x2100

Router1# configure terminal
Router1(config)# config-register 0x210F
Router1(config)# end
Router1# reload
```

Setting Environment Variables

Because many platforms can boot images from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images that the router is to use. In addition, Class A Flash file systems can load configuration files from several locations and use an environment variable to specify startup configurations.

These special environment variables are as follows:

- BOOT Environment Variable
- BOOTLDR Environment Variable
- CONFIG_FILE Environment Variable

BOOT Environment Variable

The BOOT environment variable specifies a list of bootable system images on various file systems. 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*. After you save the BOOT environment variable to your startup configuration, the router checks the variable upon startup to determine the device and filename of the image to boot.

The router tries to boot the first image in the BOOT environment variable list. If the router is unsuccessful at booting that image, it tries to boot the next image specified in the list. The router tries each image in the list until it successfully boots. If the router cannot boot any image in the BOOT environment variable list, the router attempts to boot the boot image.

If an entry in the BOOT environment variable list does not specify a device, the router assumes the device is **tftp**. If an entry in the BOOT environment variable list specifies an invalid device, the router skips that entry.

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and filename containing the boot image that the ROM monitor uses if it cannot find a valid system image. In addition, a boot image is required to boot the router with an image from a network server.

You can change the BOOTLDR environment variable on platforms that use a software boot image rather than boot ROMs. On these platforms, the boot image can be changed without having to replace the boot ROM.

This environment variable allows you to have several boot images. After you save the BOOTLDR environment variable to your startup configuration, the router checks the variable upon startup to determine which boot image to use if the system cannot be loaded.

Note Refer to your platform documentation for information on the default location of the boot image.

CONFIG_FILE Environment Variable

For Class A Flash file systems, the CONFIG_FILE environment variable specifies the file system and filename of the configuration file to use for initialization (startup). Valid file systems can include **nvram:**, **bootflash:**, **slot0:**, and **slot1:**. Refer to the "Location of Configuration Files" section in the "Modifying, Downloading, and Maintaining Configuration Files" chapter for more information on devices. After you save the CONFIG_FILE environment variable to your startup configuration, the router checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The router uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the router detects a problem with NVRAM or a checksum error, the router enters **setup** mode. Refer to the "Using Setup for Configuration Changes" chapter in this publication for more information on the **setup** command facility.

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, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** 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. Refer to the "Specify the Startup Configuration File" section in the "Modifying, Downloading, and Maintaining Configuration Files" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the CONFIG_FILE variable.

Note When you use these three 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. Use 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, BOOTLDR, and the CONFIG_FILE environment variables by issuing the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration as well as in the running configuration if a running configuration setting differs from a startup configuration setting.

Use the **more nvram:startup-config** command to display the contents of the configuration file pointed to by the CONFIG_FILE environment variable.

Setting the BOOTLDR Environment Variable

To set the BOOTLDR environment variable, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**dir** [*flash-filesystem*:]	Verifies that internal Flash or bootflash contains the boot helper image.
Step 2	**configure terminal**	Enters the configuration mode from the terminal.
Step 3	**boot bootldr** *file-url*	Sets the BOOTLDR environment variable to specify the Flash device and filename of the boot helper image. This step modifies the runtime BOOTLDR environment variable.
Step 4	**end**	Exits configuration mode.
Step 5	**copy system:running-config nvram:startup-config**	Saves this runtime BOOTLDR environment variable to your startup configuration.
Step 6	**show bootvar**	(Optional) Verifies the contents of the BOOTLDR environment variable.

The following example sets the BOOTLDR environment to change the location of the boot helper image from internal Flash to slot 0.

```
Router# dir bootflash:
-#- -length- -----date/time------ name
1   620      May 04 1995 26:22:04 rsp-boot-m
2   620      May 24 1995 21:38:14 config2

7993896 bytes available (1496 bytes used)
Router# configure terminal
Router (config)# boot bootldr slot0:rsp-boot-m
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

```
[ok]
Router# show bootvar
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:router-config

Configuration register is 0x0
```

Scheduling a Reload of the System Image

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

Note A scheduled reload must take place within approximately 24 days.

Configuring a Scheduled Reload

To configure the router to reload the Cisco IOS software at a later time, use one of the following commands in privileged EXEC command mode:

Command	Purpose
reload in [*hh*:]*mm* [*text*]	Schedules a reload of the software to take effect in *mm* minutes (or *hh* hours and *mm* minutes) from now.
reload at *hh*:*mm* [*month day* | *day month*] [*text*]	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 The **at** keyword can only be used if the system clock has been set on the router (either through NTP, the hardware calendar, or manually). The time is relative to the configured time zone on the router. To schedule reloads across several routers to occur simultaneously, the time on each router must be synchronized with NTP.

The following example illustrates how to use the **reload** command to reload the software on the router on the current day at 7:30 p.m.:

```
Router# 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]
```

The following example illustrates how to use the **reload** command to reload the software on the router at a future time:

```
Router# 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]
```

Display Information about a Scheduled Reload

To display information about a previously scheduled reload or to determine if a reload has been scheduled on the router, use the following command in EXEC command mode:

Command	Purpose
`show reload`	Display 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.

Cancel a Scheduled Reload

To cancel a previously scheduled reload, use the following command in privileged EXEC command mode:

Command	Purpose
`reload cancel`	Cancel a previously scheduled reload of the software.

The following example illustrates how to use the **reload cancel** command to stop a scheduled reload:

```
Router# reload cancel
Router#
***
*** --- SHUTDOWN ABORTED ---
***
```

Entering ROM Monitor Mode

During the first 60 seconds of startup, you can force the router to stop booting. The router will enter ROM Monitor mode, where you can change the configuration register value or boot the router manually.

To stop booting and enter ROM monitor mode, use the following commands in EXEC mode:

	Command	Purpose

Step 1	`reload`
	
	`Press the Break`[1] `key during the first 60 seconds while the system is booting.`
		Enter ROM monitor mode from privileged EXEC mode.

Step 2	`?`	List the ROM monitor commands.

1. This key will not work on the Cisco 7000 unless it has at least Cisco IOS Release 10 boot ROMs.

Timesaver If you are planning to use ROM monitor mode on a regular basis, or wish users to load using ROM monitor commands, you can configure the system to default to ROMMON. To automatically boot your system in ROM monitor mode, reset the configuration register to 0x0 by using the **config-register 0x0** configuration command. The new configuration register value, 0x0, takes effect after the router or access server is rebooted with the **reload** command. If you set the configuration to 0x0, you will have to manually boot the system from the console each time you reload the router or access server.

To exit ROMMON mode, use the continue command. If you have changed the configuration, use the **copy running-config startup-config** command and then issue the **reload** command to save your configuration changes.

Aliasing ROM Monitoring Commands

The ROM monitor supports command aliasing modeled on the aliasing function built into the Korn shell. The alias command is used to set and view aliased names. This allows the user to alias command names to a letter or word. Aliasing is often used to shorten command names or automatically invoke command options.

Aliases are stored in NVRAM and remain intact across periods of no power. These are some of the set aliases:

- b=boot
- h=history
- i=intialize/reset
- r=repeat
- k=stack
- ?=help

The following example shows a pre-aliased menu-type list for ROMMON commands:

```
> ?
$ state      Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
             Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V   Deposit value V of size S into location L with modifier M
E /S M L     Examine location L with size S with modifier M
G [address]  Begin execution
H            Help for commands
I            Initialize
K            Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
             Load system image from ROM or from TFTP server, but do not
             begin execution
O            Show configuration register option settings
P            Set the break point
S            Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

If your options appear in the above menu-type format, you can use the listed aliased commands. To initialize the router or access server, enter the **i** command. The **i** command causes the bootstrap program to reinitialize the hardware, clear the contents of memory, and boot the system. To boot the system image file, use the **b** command.

The ROM Monitor software characteristics will vary depending on your platform. For further details on ROM monitor mode commands, refer to the appropriate hardware installation guide, or perform a search on CCO.

Manually Loading a System Image from ROM Monitor

If your router does not find a valid system image, or if its configuration file is corrupted at startup, or the configuration register is set to enter ROM monitor mode, the system enters ROM monitor mode. From this mode, you can manually load a system image from the following locations:

- Internal Flash memory or a Flash memory PC card
- A network server file
- ROM
- A local or remote computer, using the Xmodem or Ymodem protocol (Cisco 1600 series and Cisco 3600 series only)

You may only boot from a location if the router can store an image there. Therefore, not all platforms can manually load from these locations.

You can also enter ROM monitor mode by restarting the router and then pressing the **Break** key or issuing a "send break" command from a telnet session during the first 60 seconds of startup.

Manually Booting from Flash Memory in ROMMON

To manually boot from Flash memory, use the following command in ROM Monitor mode:

Command	Purpose
boot flash [*filename*] **boot flash** *partition-number*:[*filename*] **boot flash flash:**[*partition-number*:] [*filename*] **boot** [*flash-fs*:][*partition-number*:][*filename*] (Cisco 1600 series and Cisco 3600 series) **boot** *device*:[*filename*] (Cisco 7000 family)	Manually boot the router from Flash. Refer to your hardware documentation for the correct form of this command to use.

If the filename is not specified, the first bootable file found in the device and partition is used.

In the following example, a router is manually booted from Flash memory. Because the optional *filename* argument is absent, the first valid file in Flash memory is loaded.

```
> boot flash
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK -
1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
```

In the following example, the **boot flash** command is used with the filename gs7-k—the name of the file that is loaded:

```
> boot flash gs7-k
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
System Bootstrap, Version 4.6(1012) [mlw 99], INTERIM SOFTWARE
Copyright (c) 1986-1992 by cisco Systems
RP1 processor with 16384 Kbytes of memory
```

The following command instructs the ROM monitor to boot the first file in the first partition of internal Flash memory:

```
> boot flash:
```

This command instructs the ROM monitor to boot the first file in the second partition of the Flash memory card in slot 0:

```
> boot slot0:2:
```

In this example, the ROM monitor boots the file named image name from the third partition of the Flash memory card in slot 0:

```
> boot slot0:3:imagename
```

The following command fails to specify a valid device type (**flash:**, **slot0:**, or **slot1:**), so the ROM monitor invokes the boot helper to boot a system image.

```
> boot flash
```

Manually Booting from a Network File in ROMMON

To manually boot from a network file, use the following command in ROM Monitor mode:

Command	Purpose
boot *filename* [*ip-address*]	Manually boots the router from a network file.

In the following example, a router is manually booted from the network file *network1*:

```
>boot network1
```

Manually Booting from ROM in ROMMON

To manually boot the router from ROM, use the following command in ROM Monitor mode:

Command	Purpose
boot	Manually boots the router from ROM.

On the Cisco 7200 series and Cisco 7500 series, the **boot** command loads the first bootable image located in bootflash.

In the following example, a router is manually booted from ROM:

```
>boot
```

Manually Booting Using MOP in ROMMON

You can interactively boot system software using MOP. Typically, you do this to verify that system software has been properly installed on the MOP boot server before configuring the router to automatically boot the system software image.

To manually boot the router using MOP, use the following command in ROM Monitor mode:

Command	Purpose
boot system mop *filename* [*mac-address*] [*interface*]	Manually boots the router using MOP.

The Cisco 7200 series and Cisco 7500 series do not support the **boot mop** command.

In the following example, a router is manually booted from a MOP server:

```
>boot mop network1
```

Exiting from ROMMON

To return to EXEC mode from the ROM monitor, you must continue loading from the default system image. To exit ROMMON mode and resume loading, use the following command in ROM monitor mode:

Command	Purpose
`continue`	Return to EXEC mode to use the system image.

Configuring High System Availability on the Cisco 7500 Series

High system availability (HSA) refers to how quickly your router returns to an operational status after a failure occurs. On the Cisco 7507 and Cisco 7513, you can install two RSP cards in a single router to improve system availability.

To configure HSA operation, you must have a Cisco 7507 or Cisco 7513 containing two RSP processor cards. The Cisco 7505 and Cisco 7576 do not support the HSA feature. For HSA compatibility, download a Cisco IOS software subset image that has a "v" in it. For example, rsp-jv-mz, rsp-ajv-mz, and rsp-pv-mz are all HSA-compatible Cisco IOS subset images.

Two RSP cards in a router provide the most basic level of increased system availability through a "cold restart" feature. A "cold restart" means that when one RSP card fails, the other RSP card reboots the router. In this way, your router is never in a failed state for very long, thereby increasing system availability.

- When one RSP card takes over operation from another, system operation is interrupted. This change is similar to issuing the **reload** command. The following events occur when one RSP card fails and the other takes over:
- The router stops passing traffic.
- Route information is lost.
- All connections are lost.
- The backup or "slave" RSP card becomes the active or "master" RSP card that reboots and runs the router. Thus, the slave has its own image and configuration file so that it can act as a single processor.

Note HSA does not impact performance in terms of packets per second or overall bandwidth. Additionally, HSA does not provide fault-tolerance or redundancy.

Note Boot ROM revision 11.1(2) or higher is required for HSA to work with an RSP2 line card.

The boot ROM is on a SIMM on the RSP2 and cannot be upgraded. You can identify the boot ROM version on your RSP2 by issuing the **show version | begin ROM** command in privileged EXEC mode.

Understanding Master and Slave Operation

A router configured for HSA operation has one RSP card that is the master and one that is the slave. The master RSP card functions as if it were a single processor, controlling all functions of the router. The slave RSP card does nothing but actively monitor the master for failure.

A system crash can cause the master RSP to fail or go into a nonfunctional state. When the slave RSP detects a nonfunctional master, the slave resets itself and takes part in *master-slave arbitration*. Master-slave arbitration is a ROM monitor process that determines which RSP card is the master and which is the slave upon startup (or reboot).

If a system crash causes the master RSP to fail, the slave RSP becomes the new master RSP and uses its own system image and configuration file to reboot the router. The failed RSP card now becomes the slave. The failure state of the slave (formerly the master) can be accessed from the console via the **show stacks** command.

With HSA operation, the following items are important to note:

- An RSP card that acts as the slave runs a different software version than it does when it acts as the master. The slave mode software is a subset of the master mode software.
- The two RSP cards do not have to run the same master software image and configuration file. When the slave reboots the system and becomes the new master, it uses its own system image and configuration file to reboot the router.
- When enabled, automatic synchronization mode automatically ensures that the master and slave RSP card have the same configuration file.
- Both hardware and software failures can cause the master RSP to enter a nonfunctional state; but, the system does not indicate the type of failure.
- The console is always connected to master. A Y cable is shipped with your Cisco 7507 or Cisco 7513. The “top” of the Y cable plugs into the console port on each RSP card, while the “bottom” of the Y cable plugs into a terminal or terminal server. The master RSP card has ownership of the Y cable in that the slave Universal Asynchronous Receiver Transmitter (UART) drivers are disabled. Thus, no matter which RSP card has mastership of the system, your view of the internetwork environment is always from the master’s perspective. Refer to your product’s hardware installation and maintenance publication for information on properly installing the Y cable.

Understanding HSA Implementation Methods

There are two common ways to use HSA. You can use HSA for:

- Simple hardware backup

  Use this method to protect against an RSP card failure. With this method, you configure both RSP cards with the same software image and configuration information. Also, you configure the router to automatically synchronize configuration information on both cards when changes occur.

- Software error protection

  Use this method to protect against critical Cisco IOS software errors in a particular release. With this method, you configure the RSP cards with different software images, but with the same configuration information. If you are using new or experimental Cisco IOS software, consider using the software error protection method.

You can also use HSA for advanced implementations. For example, you can configure the RSP cards with the following:

- Similar software versions, but different configuration files
- Different software images *and* different configuration files
- Widely varied configuration files (for example, various features or interfaces can be turned off and on per card)

Note While other uses are possible, the configuration information in this guide describes commands for only the two common methods—simple hardware backup and software error protection.

HSA Configuration Task List

When configuring HSA operation, complete the tasks in the following sections. The first two and last two tasks are required for both implementations. The third and fourth tasks relates to simple hardware backup. The fifth task relates to software error protection only.

- Specifying the Default Slave RSP (both implementations)
- Ensuring That Both RSP Cards Contain the Same Configuration File (both implementations)
- Ensuring That Both RSP Cards Contain the Same System Image (simple hardware backup only)
- Ensuring That Both RSP Cards Contain the Same Microcode Image (simple hardware backup only)
- Specifying Different Startup Images for the Master and Slave RSP (software error protection only)
- Setting Environment Variables on the Master and Slave RSP (both implementations)
- Monitoring and Maintaining HSA Operation (both implementations)

Specifying the Default Slave RSP

Because your view of the environment is always from the master RSP perspective, you define a default slave RSP. The router uses the default slave information when booting as follows:

- If a system boot is due to powering up the router or using the **reload** command, then the specified default slave will be the slave RSP.
- If a system boot is due to a system crash or hardware failure, then the system ignores the default slave designation and makes the crashed or faulty RSP the slave RSP.

To define the default slave RSP, use the following command, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`configure terminal`	Enter the configuration mode from the terminal.
Step 2	`slave default-slot` *`processor-slot-number`*	Define the default slave RSP.
Step 3	`end`	Exit configuration mode.
Step 4	`copy system:running-config nvram:startup-config`	Save this information to your startup configuration.

Upon the next system reboot, the above changes take effect (if both RSP cards are operational). Thus, the specified default slave becomes the slave RSP card. The other RSP card takes over mastership of the system and controls all functions of the router.

If you do not specifically define the default slave RSP, the RSP card located in the higher number processor slot is the default slave. On the Cisco 7507, processor slot 3 contains the default slave RSP. On the Cisco 7513, processor slot 7 contains the default slave RSP.

The following example sets the default slave RSP to processor slot 2 on a Cisco 7507:

```
Router# configure terminal
Router (config)# slave default-slot 2
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Ensuring That Both RSP Cards Contain the Same Configuration File

With both the simple hardware backup and software error protection implementation methods, you always want your master and slave configuration files to match. To ensure that they match, turn on automatic synchronization. In automatic synchronization mode, the master copies its startup configuration to the slave's startup configuration when you issue a **copy** command that specifies the master's startup configuration (**nvram:startup-config**) as the target.

Automatic synchronization mode is on by default; however, to turn it on manually, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`configure terminal`	Enter the configuration mode from the terminal.
Step 2	`slave auto-sync config`	Turn on automatic synchronization mode.
Step 3	`end`	Exit configuration mode.
Step 4	`copy system:running-config nvram:startup-config`	Save this information to your startup configuration and copy the configuration to the slave's startup configuration.

The following example turns on automatic configuration file synchronization:

```
Router# configure terminal
Router (config)# slave auto-sync config
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Ensuring That Both RSP Cards Contain the Same System Image

For simple hardware backup, ensure that both RSP cards have the same system image.

To ensure that both RSP cards have the same system image, use the following commands in EXEC mode:

	Command	Purpose
Step 1	**show bootvar**	Display the contents of the BOOT environment variable to learn the current booting parameters for the master and slave RSP.
Step 2	**dir** {**bootflash:** | **slot0:** | **slot1:**}	Verify the location and version of the master RSP software image.
Step 3	**dir** {**slavebootflash:** | **slaveslot0:** | **slaveslot1:**}	Determine if the slave RSP contains the same software image in the same location.
Step 4	**copy** {**bootflash:**[*filename*] | **slot0:**[*filename*] | **slot1:**[*filename*]}{**slavebootflash:**[*filename*] | **slaveslot0:**[*filename*] | **slaveslot1:**[*filename*]} Note that you might also have to use the **delete** and/or **squeeze** command in conjunction with the **copy** command to accomplish this step.	If the slave RSP does not contain the same system image in the same location, copy the master's system image to the appropriate slave location.

The following example shows the process of ensuring that both RSP cards have the same system image. Note that because no environment variables are set, the default environment variables are in effect for both the master and slave RSP. Therefore, the router will boot the image in slot 0.

```
Router# show bootvar

BOOT variable =
CONFIG_FILE variable =
Current CONFIG_FILE variable =
BOOTLDR variable does not exist

Configuration register is 0x0

current slave is in slot 7
BOOT variable =
CONFIG_FILE variable =
BOOTLDR variable does not exist

Configuration register is 0x0

Router# dir slot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2

7993896 bytes available (1496 bytes used)

Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)

Router# delete slaveslot0:rsp-k-mz11.1
Router# copy slot0:rsp-k-mz11.2 slaveslot0:rsp-k-mz11.2
```

Ensuring That Both RSP Cards Contain the Same Microcode Image

To ensure that interface processors will load the same microcode, regardless of which RSP is used, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**show controller cbus**	Determine the microcode images used on the interface processors. If all interface processors are running from the bundled system microcode, no further action is required.
Step 2	**dir** {**bootflash:** | **slot0:** | **slot1:**}	If any interface processors are running from the flash file system, verify the location and version of the master RSP's supplementary microcode.
Step 3	**dir** {**slavebootflash:** | **slaveslot0:** | **slaveslot1:**}	Determine if the slave RSP contains the same microcode image in the same location.
Step 4	**copy** {**bootflash:**[*filename*] | **slot0:**[*filename*] | **slot1:**[*filename*]} {**slavebootflash:**[*filename*] | **slaveslot0:**[*filename*] | **slaveslot1:**[*filename*]} Note that you might also have to use the **delete** and/or **squeeze** command in conjunction with the **copy** command to accomplish this step.	If the slave RSP does not contain the same microcode image in the same location, copy the master's microcode image to the appropriate slave location.

The following example ensures that both RSP cards have the same microcode image. Notice that slots 0, 1, 4, 9, and 10 load microcode from the bundled software, as noted by the statement "software loaded from system." Slot 11, the (Fast Serial Interface Processor) FSIP processor, does not use the microcode bundled with the system. Instead, it loads the microcode from slot0:pond/bath/rsp_fsip20-1. Thus, you must ensure that the slave RSP has a copy of the same FSIP microcode in the same location.

```
Router# show controller cbus

MEMD at 40000000, 2097152 bytes (unused 416, recarves 3, lost 0)
  RawQ 48000100, ReturnQ 48000108, EventQ 48000110
  BufhdrQ 48000128 (2948 items), LovltrQ 48000140 (5 items, 1632 bytes)
  IpcbufQ 48000148 (16 items, 4096 bytes)
  3571 buffer headers (48002000 - 4800FF20)
  pool0: 28 buffers, 256 bytes, queue 48000130
  pool1: 237 buffers, 1536 bytes, queue 48000138
  pool2: 333 buffers, 4544 bytes, queue 48000150
  pool3: 4 buffers, 4576 bytes, queue 48000158
  slot0: EIP, hw 1.5, sw 20.00, ccb 5800FF30, cmdq 48000080, vps 4096
    software loaded from system
    Ethernet0/0, addr 0000.0ca3.cc00 (bia 0000.0ca3.cc00)
      gfreeq 48000138, lfreeq 48000160 (1536 bytes), throttled 0
      rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 2
      txq 48000168, txacc 48000082 (value 27), txlimit 27
                  .........
slot1: FIP, hw 2.9, sw 20.02, ccb 5800FF40, cmdq 48000088, vps 4096
    software loaded from system
    Fddi1/0, addr 0000.0ca3.cc20 (bia 0000.0ca3.cc20)
      gfreeq 48000150, lfreeq 480001C0 (4544 bytes), throttled 0
      rxlo 4, rxhi 165, rxcurr 0, maxrxcurr 0
      txq 480001C8, txacc 480000B2 (value 0), txlimit 95
 slot4: AIP, hw 1.3, sw 20.02, ccb 5800FF70, cmdq 480000A0, vps 8192
    software loaded from system
    ATM4/0, applique is SONET (155Mbps)
```

```
     gfreeq 48000150, lfreeq 480001D0 (4544 bytes), throttled 0
     rxlo 4, rxhi 165, rxcurr 0, maxrxcurr 0
     txq 480001D8, txacc 480000BA (value 0), txlimit 95
slot9: MIP, hw 1.0, sw 20.02, ccb 5800FFC0, cmdq 480000C8, vps 8192
   software loaded from system
   T1 9/0, applique is Channelized T1
     gfreeq 48000138, lfreeq 480001E0 (1536 bytes), throttled 0
     rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 0
     txq 480001E8, txacc 480000C2 (value 27), txlimit 27
               .......

slot10: TRIP, hw 1.1, sw 20.00, ccb 5800FFD0, cmdq 480000D0, vps 4096
   software loaded from system
   TokenRing10/0, addr 0000.0ca3.cd40 (bia 0000.0ca3.cd40)
     gfreeq 48000150, lfreeq 48000200 (4544 bytes), throttled 0
     rxlo 4, rxhi 165, rxcurr 1, maxrxcurr 1
     txq 48000208, txacc 480000D2 (value 95), txlimit 95
              .........

slot11: FSIP, hw 1.1, sw 20.01, ccb 5800FFE0, cmdq 480000D8, vps 8192
   software loaded from flash slot0:pond/bath/rsp_fsip20-1
   Serial11/0, applique is Universal (cable unattached)
     gfreeq 48000138, lfreeq 48000240 (1536 bytes), throttled 0
     rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 0
     txq 48000248, txacc 480000F2 (value 5), txlimit 27
             ...........

Router# dir slot0:pond/bath/rsp_fsip20-1
-#- -length- -----date/time------ name
3   10242    Jan 01 1995 03:46:31 pond/bath/rsp_fsip20-1

Router# dir slaveslot0:pond/bath/rsp_fsip20-1
No such file

4079832 bytes available (3915560 bytes used)

Router# copy slot0:pond/bath/rsp_fsip20-1 slaveslot0:
4079704 bytes available on device slaveslot0, proceed? [confirm]

Router# dir slaveslot0:pond/bath/rsp_fsip20-1
-#- -length- -----date/time------ name
3   10242    Mar 01 1993 02:35:04 pond/bath/rsp_fsip20-1

4069460 bytes available (3925932 bytes used)
```

Specifying Different Startup Images for the Master and Slave RSP

For software error protection, the RSP cards should have different system images.

When the factory sends you a new Cisco 7507 or Cisco 7513 with two RSPs, you receive the same system image on both RSP cards. For the software error protection method, you need two different software images on the RSP cards. Thus, you copy a desired image to the master RSP card and modify the **boot system** commands to reflect booting two different system images. Each RSP card uses its own image to boot the router when it becomes the master.

To specify different startup images for the master and slave RSP, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	**dir** {**bootflash:** | **slot0:** | **slot1:**}	Verify the location and version of the master RSP software image.
Step 2	**dir** {**slavebootflash:** | **slaveslot0:** | **slaveslot1:**}	Determine if the slave RSP contains the same software image in the same location.
Step 3	**copy** *source-url* {**bootflash:** | **slot0:** | **slot1:**}	Copy a different system image to the master RSP.
Step 4	**configure terminal**	Enter configuration mode from the terminal.
Step 5	**boot system flash bootflash:**[*filename*] **boot system flash slot0:**[*filename*] **boot system flash slot1:**[*filename*]	From global configuration mode, configure the master RSP to boot the new image from the appropriate location.
Step 6	**boot system flash bootflash:**[*filename*] **boot system flash slot0:**[*filename*] **boot system flash slot1:**[*filename*]	Also, add a **boot system** command that specifies the slave's boot image and location. This is the boot image that the slave uses when it becomes the master RSP and boots the system. Note that the because the slave will boot this image when the slave is actually the new master RSP, the command syntax does not use a "**slave**" prefix.
Step 7	**boot system** {**rcp** | **tftp** | **ftp**} [*filename*] [*ip-address*]	(Optional) Configure the master RSP to boot from a network server.
Step 8	**config-register** *value*[1]	Set the configuration register to enable the system to load the system image from a network server or from Flash.
Step 9	**end**	Exit configuration mode.
Step 10	**copy system:running-config nvram:startup-config**	Save the configuration file to the master's startup configuration. Because automatic synchronization is turned on, this step saves the **boot system** commands to the master and slave startup configuration.
Step 11	**reload**	Reset the router with the new configuration information.

1. Refer to the "Modifying the Configuration Register Boot Field" section for more information on systems that can use this command to modify the software configuration register.

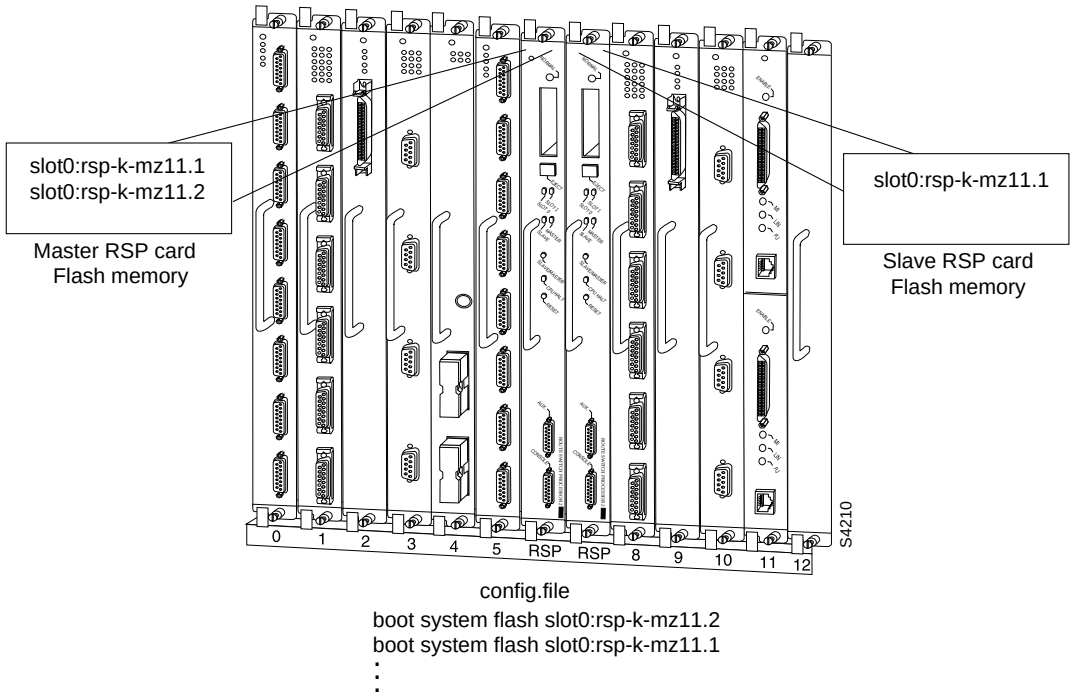
HSA: Upgrading to a New Software Version Example

In this example, assume the following:

- The master RSP is in processor slot 6 and the slave RSP is in processor slot 7 of a Cisco 7513.
- The system has the same image rsp-k-mz11.1 in PCMCIA slot 0 of both the master and slave RSP card.
- You want to upgrade to Cisco IOS Release 12.0, but you want to guard against software failures. So, you configure HSA operation for software error protection.

Figure 12 illustrates the software error protection configuration for this example. The configuration commands for this configuration follow the figure.

Figure 12 Software Error Protection: Upgrading to a New Software Version



Because you always view the environment from the master RSP perspective, in the following command you view the master's slot 0 to verify the location and version of the master's software image:

```
Router# dir slot0:
-#- -length- -----date/time------ name
1    3482496  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)
```

Now view the slave's software image location and version:

```
Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482496  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)
```

Because you want to run the Release 12.0 system image on one RSP card and the Release 11.1 system image on the other RSP card, copy the Release 12.0 system image to the master's slot 0:

```
Router# copy tftp: slot0:rsp-k-mz12.0
```

Enter global configuration mode and configure the system to boot first from a Release 12.0 system image and then from a Release 11.1 system image.

```
Router# configure terminal
Router (config)# boot system flash slot0:rsp-k-mz12.0
Router (config)# boot system flash slot0:rsp-k-mz11.1
```

With this configuration, when the slot 6 RSP card is master, it looks first in its PCMCIA slot 0 for the system image file rsp-k-mz11.2 to boot. Finding this file, the router boots from that system image. When the slot 7 RSP card is master, it also looks first in its slot 0 for the system image file rsp-k-mz12.0 to boot. Because that image does not exist in that location, the slot 7 RSP card looks for the system image file rsp-k-mz11.1 in slot 0 to boot. Finding this file in its PCMCIA slot 0, the router boots from that system image. In this way, each RSP card can reboot the system using its own system image when it becomes the master RSP card.

Configure the system further with a fault-tolerant booting strategy:

```
Router (config)# boot system tftp rsp-k-mz11.1 192.168.1.25
```

Set the configuration register to enable loading of the system image from a network server or from Flash and save the changes to the master and slave startup configuration file:

```
Router (config)# config-register 0x010F
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Reload the system so that the master RSP uses the new Release 12.0 system image:

```
Router# reload
```

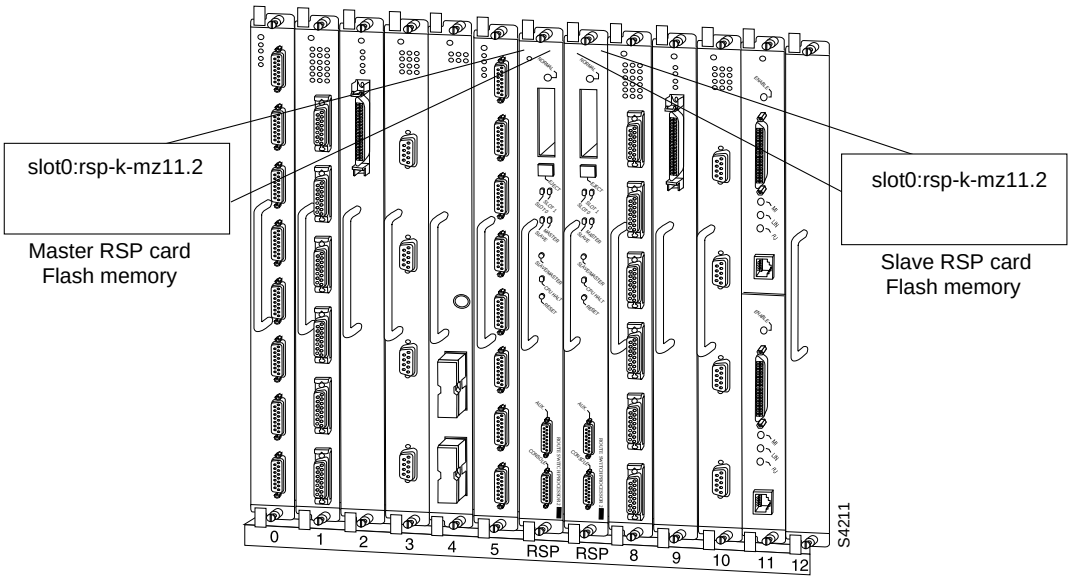
HSA: Backing Up with an Older Software Version Example

In the following example, assume the following:

- The master RSP is in processor slot 6 and the slave RSP is in processor slot 7 of a Cisco 7513.
- The system has the same image rsp-k-mz11.2 in PCMCIA slot 0 of both the master and slave RSP card.
- You want to use to Cisco IOS Release 11.1 as backup to guard against software failures. So, you configure HSA operation for software error protection.

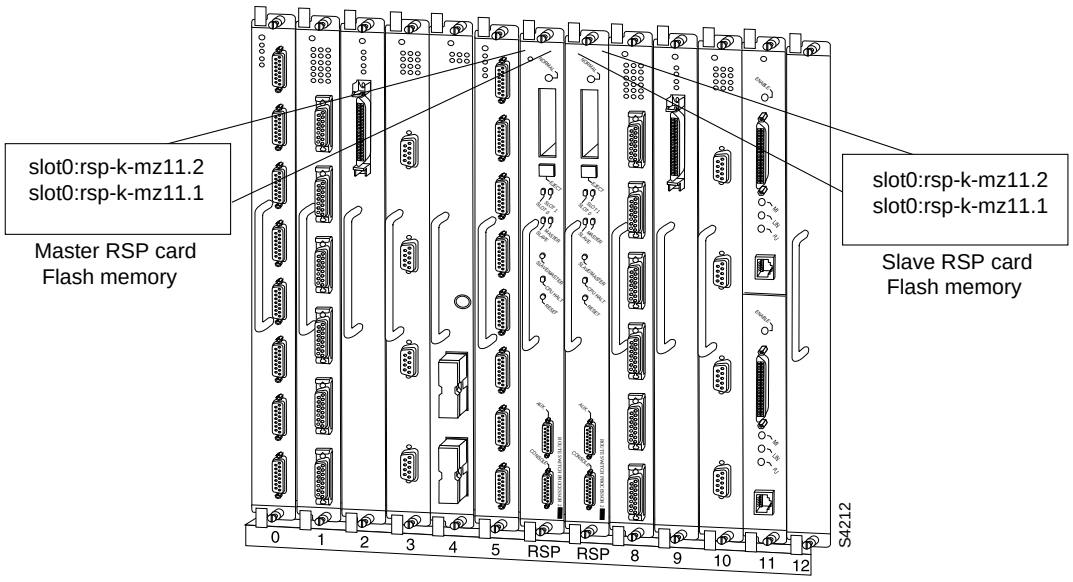
In this scenario, you begin with the configuration shown in Figure 13.

Figure 13 Software Error Protection: Backing Up with an Older Software Version, Part I



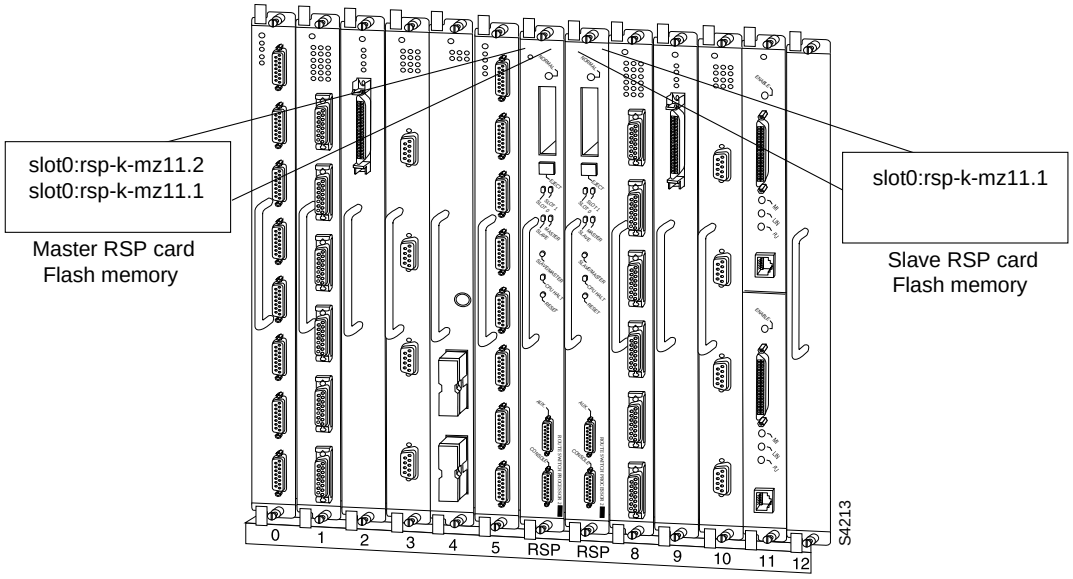
First, copy the rsp-k-mz11.1 image to the master and slave RSP card, as shown in Figure 14.

Figure 14 Software Error Protection: Backing Up with an Older Software Version, Part II



Next, you delete the rsp-k-mz11.2 image from the slave RSP card. The final configuration is shown in Figure 15.

Figure 15 Software Error Protection: Backing Up with an Older Software Version, Part III



The following commands configure software error protection for this example scenario.

View the master and slave slot 0 to verify the location and version of their software images:

```
Router# dir slot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2

7993896 bytes available (1496 bytes used)

Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2

7993896 bytes available (1496 bytes used)
```

Copy the Release 11.1 system image to the master and slave slot 0:

```
Router# copy tftp: slot0:rsp-k-mz11.1
Router# copy tftp: slaveslot0:rsp-k-mz11.1
```

Delete the rsp-k-mz11.2 image from the slave RSP card:

```
Router# delete slaveslot0:rsp-k-mz11.2
```

Configure the system to boot first from a Release 11.2 system image and then from a Release 11.1 system image:

```
Router# configure terminal
Router (config)# boot system flash slot0:rsp-k-mz11.2
Router (config)# boot system flash slot0:rsp-k-mz11.1
```

Configure the system further with a fault-tolerant booting strategy:

```
Router (config)# boot system tftp rsp-k-mz11.1 192.168.1.25
```

Set the configuration register to enable loading of the system image from a network server or from Flash and save the changes to the master and slave startup configuration file:

```
Router (config)# config-register 0x010F
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Note You do not need to reload the router in this example, because the router is currently running the Release 11.2 image.

Setting Environment Variables on the Master and Slave RSP

You can optionally set environment variables on both RSP cards in a Cisco 7507 and Cisco 7513. For more information on environment variables, refer to the “Setting Environment Variables” section.

Note When configuring HSA operation, Cisco recommends that you use the default environment variables. If you change the variables, Cisco recommends setting the same device for equivalent environment variables on each RSP card. For example, if you set one RSP card’s CONFIG_FILE environment variable device to NVRAM, set the other RSP card’s CONFIG_FILE environment variable device to NVRAM also.

You set environment variables on the master RSP just as you would if it were the only RSP card in the system. Refer to the following sections for more information on these steps:

- "Controlling Environment Variables" section on page FC-217
- "Specifying the Startup Configuration File" section on page FC-155 (in the "Loading and Maintaining System Images and Microcode" chapter)
- "Setting the BOOTLDR Environment Variable" section on page FC-218
- "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section on page FC-155 (in the "Modifying, Downloading, and Maintaining Configuration Files" chapter)

You can set the same environment variables on the slave RSP card, manually or automatically. The following sections describe these two methods:

- Automatically Setting Environment Variables on the Slave RSP
- Manually Setting Environment Variables on the Slave RSP

Automatically Setting Environment Variables on the Slave RSP

With automatic synchronization turned on, the system automatically saves the same environment variables to the slave's startup configuration when you set the master's environment variables and save them.

Note Automatic synchronization mode is on by default. To turn off automatic synchronization, use the **no slave auto-sync config** global configuration command.

To set environment variables on the slave RSP when automatic synchronization is on, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1		Set the master's environment variables as described in the "Controlling Environment Variables," "Setting the BOOTLDR Environment Variable," and "Specify the CONFIG_FILE Environment Variable (Class A Flash File Systems)" sections.
Step 2	`copy system:running-config nvram:startup-config`	Save the settings to the startup configuration. This also puts the information under that RSP card's ROM monitor control.
Step 3	`show bootvar`	Verify the environment variable settings.

Manually Setting Environment Variables on the Slave RSP

If you disable automatic synchronization of configuration files, you must manually synchronize the slave's configuration file to the master's configuration file to store environment variables on the slave RSP.

Once you set the master's environment variables, you can manually set the same environment variables on the slave RSP card using the **slave sync config** command.

To manually set environment variables on the slave RSP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1		Set the master's environment variables as described in the "Controlling Environment Variables," "Setting the BOOTLDR Environment Variable," and "CONFIG_FILE Environment Variable" sections.
Step 2	**end**	Exit global configuration mode.
Step 3	**copy system:running-config nvram:startup-config**	Save the settings to the startup configuration. This also puts the information under that RSP card's ROM monitor control.
Step 4	**slave sync config**	Save the same environment variables to the slave RSP by manually synchronizing their configuration files.
Step 5	**show bootvar**	Verify the environment variable settings.

Monitoring and Maintaining HSA Operation

To monitor and maintain HSA operation, complete the following tasks in the following sections:

- Overriding the Slave Image Bundled with the Master Image
- Manually Synchronizing Configuration Files
- Troubleshooting a Failed RSP Card
- Disabling Access to Slave Console
- Displaying Information About Master and Slave RSP Cards

Overriding the Slave Image Bundled with the Master Image

You can override the slave image that is bundled with the master image. To do so, use the following command in global configuration mode:

Command	Purpose
slave image {**system** | *file-url*}	Specify which image the slave runs.

Manually Synchronizing Configuration Files

You can manually synchronize configuration files and ROM monitor environment variables on the master and slave RSP card. To do so, use the following command in privileged EXEC mode:

Command	Purpose
slave sync config	Manually synchronize master and slave configuration files.

Caution When you install a second RSP card for the first time, you *must* immediately configure it using the **slave sync config** command. This ensures that the new slave is configured consistently with the master. Failure to do so can result in an unconfigured slave RSP card taking over mastership of the router when the master fails, rendering the network inoperable.

The **slave sync config** command is also a useful tool for more advanced implementation methods not discussed in this chapter.

Troubleshooting a Failed RSP Card

When a new master RSP card takes over mastership of the router, it automatically reboots the failed RSP card as the slave RSP card. You can access the state of the failed RSP card in the form of a stack trace from the master console using the **show stacks** command.

You can also manually reload a failed, inactive RSP card from the master console. This task returns the card to the active slave state. If the master RSP fails, the slave will be able to become the master. To manually reload the inactive RSP card, use the following command in global configuration mode:

Command	Purpose
`slave reload`	Reload the inactive slave RSP card.

Disabling Access to Slave Console

The slave console does not have enable password protection. Thus, an individual connected to the slave console port can enter privileged EXEC mode and view or erase the configuration of the router. Use the **no slave terminal** command to disable slave console access and prevent security problems. When the slave console is disabled, users cannot enter commands.

If slave console access is disabled, the following message appears periodically on the slave console:

```
%%Slave terminal access is disabled. Use "slave terminal" command in master RSP
configuration mode to enable it.
```

Displaying Information About Master and Slave RSP Cards

You can also display information about both the master and slave RSP cards. To do so, use any of the following commands in EXEC mode:

Command	Purpose
`show bootvar`	Display the environment variable settings and configuration register settings for both the master and slave RSP cards.
`show file systems`	Show a list of Flash devices currently supported on the router.
`show version`	Display the software version running on the master and slave RSP card.
`show stacks`	Display the stack trace and version information of the master and slave RSP cards.