



Using the Cisco IOS File System

This chapter describes the Cisco IOS File System (IFS) feature, which provides a single interface to all the file systems a router uses, including:

- Flash memory file systems
- Network file systems (TFTP, rcp, and FTP)
- Any other endpoint for reading or writing data (such as NVRAM, the running configuration, ROM, raw system memory, system bundled microcode, Xmodem, Flash load helper log, modems, and BRI MUX interfaces).

For a complete description of the IFS commands in this chapter, refer to the “Cisco IOS File System Commands” chapter of the *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference or search online.

IFS Use and Management Task List

This chapter describes how to use the Cisco IOS File System (IFS) feature and the tasks you can perform to manage file systems related to IFS. Information about IFS and its optional file management tasks are described in the following sections:

- Understanding IFS
- Copying Files Using URLs
- Using URLs in Commands
- General File System Management
- Flash Memory File System Types
- Remote File System Management
- NVRAM File System Management
- System File System Management

Understanding IFS

This section describes the following IFS capabilities and benefits:

- File Viewing and Classification
- Platform-Independent Commands
- Minimal Prompting for Commands
- Directory Navigation and Creation

File Viewing and Classification

With IFS, all files can be viewed and classified (image, text file, and so forth), including files on remote servers. For example, you may want to determine the size and type of an image on a remote server before you copy it to ensure that it is a valid image. You can also view a configuration file on a remote server to verify that it is the correct configuration file before you load the file on the router.

Platform-Independent Commands

With IFS, the file system user interface is no longer platform specific. Commands have the same syntax, regardless of which platform is used. Thus, you can use the same commands for all of your routers.

However, not all commands are supported on all platforms and file systems. Because different types of file systems support different operations, certain commands are not available for all file systems. Platforms will support commands for the file systems they use.

Minimal Prompting for Commands

IFS minimizes the required prompting for many commands, such as the **copy** command. You can enter all of the required information in the command line, rather than having to provide information when the system prompts you for it. For example, if you want to copy a file to an FTP server, you can specify the specific location on the router of the source file, the specific location of the destination file on the FTP server, and the username and password to use when connecting to the FTP server, all on a single line. However, if you wish to have the router prompt you for the needed information, you can still enter the minimum form of the command.

Depending on the current configuration of the **file prompt** command and the type of command you entered, the router may prompt you for confirmation, even if you have provided all the information in the command. In these cases, the default value will be the value entered in the command. Press Return to confirm the values.

Directory Navigation and Creation

With IFS, you can move around to different directories and list the files in a directory. On newer platforms, you can create subdirectories in Flash memory or on a disk.

Copying Files Using URLs

The new file system interface uses Uniform Resource Locators (URLs) to specify the location of a file. URLs are commonly used to specify files or locations on the World Wide Web. However, on Cisco routers, they can now be used to specify the location of files on the router or remote file servers.

On Cisco routers, use URLs in commands to specify the location of the file or directory. For example, if you want to copy a file from one location to another, use the **copy source-url destination-url** command.

The format of URLs used by the routers can vary from the format you may be used to using. There are also a variety of formats that can be used, based on the location of the file.

This section provides the following information for copying files using URLs:

- Specifying Files on a Network Server
- Specifying Local Files
- Using URL Prefixes

Specifying Files on a Network Server

When you want to specify a file on a network server, use one of the following forms:

ftp:*[[//[username[:password]@]location]/directory]/filename*

rcp:*[[//[username@]location]/directory]/filename*

tftp:*[[//location]/directory]/filename*

The location can be an IP address or a host name. The username variable, if specified, overrides the username specified by the **ip rcmd remote-username** or **ip ftp username** commands. The password overrides the password specified by the **ip ftp password** command.

The file path (directory and filename) is specified relative to the directory used for file transfers. For example, on UNIX file servers, TFTP pathnames start in the /tftpboot directory, and rcp and FTP paths start in the home directory associated with the username.

The following example specifies the file c7200-j-mz.112-current on the TFTP server myserver.cisco.com. The file is located in the directory called /tftpboot/master.

```
tftp://myserver.cisco.com/master/c7200-j-mz.112-current
```

The following example specifies the file ship-config on the server enterprise.cisco.com. The router uses the username jeanluc and the password secret to access this server via FTP.

```
ftp://jeanluc:secret@enterprise.cisco.com/ship-config
```

Specifying Local Files

Use the following syntax to specify a file located on the router:

prefix:[directory]/filename

You can use this form to specify a file in Flash memory or NVRAM.

The following example specifies the startup configuration in NVRAM:

```
nvram:startup-config
```

The following example specifies the file backup-config in the configs directory of Flash memory:

```
flash:configs/backup-config
```

When referring to a file system instead of a file, the form is:

prefix:

This form specifies the file system itself, rather than a file in the file system. Use this form to perform commands on file systems themselves, such as listing the files in a file system or formatting the file system.

The following example specifies the first PCMCIA Flash memory card:

```
slot0:
```

Using URL Prefixes

The URL prefix specifies the file system. File system prefixes are listed in Table 10. The list of available file systems differs by platform and operation. Refer to your product documentation or use the **show file systems** command to determine which prefixes are available on your platform.

Table 10 File System Prefixes

Prefix	File System
bootflash:	Boot Flash memory.
disk0:	Rotating media.
flash:	Flash memory. This prefix is available on all platforms. For platforms that do not have a device named flash:, the prefix flash: is aliased to slot0: . Therefore, you can use the prefix flash: to refer to the main Flash memory storage area on all platforms.
flh:	Flash load helper log files.
ftp:	File Transfer Protocol (FTP) network server.
null:	Null destination for copies. You can copy a remote file to null to determine its size.
nvram:	NVRAM.
rcp:	Remote copy protocol (rcp) network server.
slavebootflash:	Internal Flash memory on a slave RSP card of a router configured for HSA.
slavenvram:	NVRAM on a slave RSP card of a router configured for HSA.
slaveslot0:	First PCMCIA card on a slave RSP card of a router configured for HSA.
slaveslot1:	Second PCMCIA card on a slave RSP card of a router configured for HSA.

Table 10 File System Prefixes (continued)

Prefix	File System
slot0:	First PCMCIA Flash memory card.
slot1:	Second PCMCIA Flash memory card.
system:	Contains the system memory, including the running configuration.
tftp:	Trivial File Transfer Protocol (TFTP) network server.
xmodem:	Obtain the file from a network machine using the Xmodem protocol.
ymodem:	Obtain the file from a network machine using the Ymodem protocol.

**Note**

MOP servers are no longer supported as file systems.

In all commands, the colon is required after the file system name. However, commands that did not require the colon previously will continue to be supported, although they will not be available in the context-sensitive help.

URL Prefix for Partitioned Devices

For partitioned devices, the prefix includes the partition number. The syntax for the prefix on a partitioned device is *device:partition-number:*.

For example, **flash:2:** refers to the second partition in Flash memory.

URL Component Lengths

Table 11 lists the maximum lengths, in characters, of the different URL components.

Table 11 URL Component Lengths

Component	Length (Number of Characters)
Prefix	31
Username	15
Password	15
Hostname	31
Directory	63
Filename	63

Using URLs in Commands

Depending on which command you are using, different file systems are available. Some file systems can only serve as a source for files, not a destination. For example, you cannot copy to another machine using Xmodem. Other operations, such as **format** and **erase**, are only supported by certain file systems on certain platforms.

The following sections provide hints for using URLs in commands:

- Determining File Systems Supporting a Command
- Using the Default File System
- Using Tab Completion
- Listing Files in a File System

Determining File Systems Supporting a Command

Use the context sensitive help to determine which file systems can be used for a particular command. In the following example, the context-sensitive help displays which file systems can be used as sources for the **copy** command. The output will vary based on the platform.

```
Router# copy ?
/erase      Erase destination file system.
bootflash:  Copy from bootflash: file system
flash:      Copy from flash: file system
ftp:        Copy from ftp: file system
null:       Copy from null: file system
nvram:      Copy from nvram: file system
rcp:        Copy from rcp: file system
system:     Copy from system: file system
tftp:       Copy from tftp: file system
```

Using the Default File System

For most commands, if no file system is specified, the file is assumed to be in the default directory, as specified by the **cd** command.

```
Router# pwd
slot0:
Router# dir
Directory of slot0:/

 1  -rw-      4720148   Aug 29 1997 17:49:36 hampton/nitro/c7200-j-mz
 2  -rw-      4767328   Oct 01 1997 18:42:53 c7200-js-mz
 5  -rw-         639   Oct 02 1997 12:09:32 foo
 7  -rw-         639   Oct 02 1997 12:37:13 the_time

20578304 bytes total (3104544 bytes free)
Router# cd nvram:
Router# dir
Directory of nvram:/

 1  -rw-         2725           <no date> startup-config
 2  ----          0           <no date> private-config
 3  -rw-         2725           <no date> underlying-config

129016 bytes total (126291 bytes free)
```

Using Tab Completion

You can use tab completion to reduce the number of characters you need to type for a command. Type the first few characters of the filename, and press the Tab key. If the characters are unique to a filename, the router will complete the filename for you. Continue entering the command as normal and press Return to execute the command.

In the following example, the router completes the filename `startup-config` because it is the only file in the `nvr` file system that starts with “s”:

```
Router# show file info nvrs<tab>
Router# show file info nvram:startup-config<Enter>
```

If you use tab completion without specifying any characters, the router uses the first file in the file system.

```
Router# show file info nvram:<tab>
Router# show file info nvram:private-config<Enter>
```

Listing Files in a File System

For many commands, you can get a listing of the files in a file system on the router by using the context-sensitive help. In the following example, the router lists the files in NVRAM:

```
Router# show file info nvram:?
nvram:private-config nvram:startup-config nvram:underlying-config
```

General File System Management

This section describes general tasks you can perform to use the different file systems.

Listing Available File Systems

Not all file systems are supported on every platform. To list the file systems available on your platform, use the following EXEC mode command:

Command	Purpose
<code>show file systems</code>	List the file systems available on your platform. This command also displays information about each file system.

Setting the Default File System

You can specify the file system or directory that the system uses as the default file system. Setting the default file system allows you to omit an optional *filesystem:* argument from related commands. For all EXEC commands that have an optional *filesystem:* argument, the system uses the file system specified by the `cd` command when you omit the optional *filesystem:* argument. For example, the `dir` command contains an optional *filesystem:* argument and displays a list of files on the file system.

To specify a default file system, use the following EXEC mode command:

Command	Purpose
<code>cd filesystem:</code>	Set a default Flash memory device.

The following example sets the default file system to the Flash memory card inserted in slot 0:

```
cd slot0:
```

Displaying the Current Default File System

To display the current default file system, as specified by the `cd` command, use the following EXEC mode command:

Command	Purpose
<code>pwd</code>	Display the current file system.

The following example shows that the default file system is slot 0:

```
Router> pwd
slot0:
```

The following example uses the `cd` command to change the default file system to system and then uses the `pwd` command to verify that the default file system was changed:

```
Router> cd system:
Router> pwd
system:
```

Displaying Information about Files on a File System

You can view a list of the contents of a file system before manipulating its contents. For example, before copying a new configuration file to Flash memory, you may want to verify that the file system does not already contain a configuration file with the same name. Similarly, before copying a Flash configuration file to another location, you may want to verify its filename for use in another command.

To show display information about files on a file system, use one of the following EXEC mode commands:

Command	Purpose
<code>dir [/all] [filesystem:] [filename]</code>	Display a list of files on a file system.
<code>show file systems</code>	Display more information about each of the files on a file system.
<code>show file information file-url</code>	Display information about a specific file.
<code>show file descriptors</code>	Display a list of open file descriptors.

The following example compares the different commands used to display information about files for the PCMCIA card in the first slot. Notice that deleted files appear in the **dir /all** output but not in the **dir** output.

```
Router# dir slot0:
Directory of slot0:/

 1  -rw-   4720148   Aug 29 1997 17:49:36 hampton/nitro/c7200-j-mz
 2  -rw-   4767328   Oct 01 1997 18:42:53 c7200-js-mz
 5  -rw-     639    Oct 02 1997 12:09:32 foo
 7  -rw-     639    Oct 02 1997 12:37:13 the_time

20578304 bytes total (3104544 bytes free)

Router# dir /all slot0:
Directory of slot0:/

 1  -rw-   4720148   Aug 29 1997 17:49:36 hampton/nitro/c7200-j-mz
 2  -rw-   4767328   Oct 01 1997 18:42:53 c7200-js-mz
 3  -rw-   7982828   Oct 01 1997 18:48:14 [rsp-jsv-mz]
 4  -rw-     639    Oct 02 1997 12:09:17 [the_time]
 5  -rw-     639    Oct 02 1997 12:09:32 foo
 6  -rw-     639    Oct 02 1997 12:37:01 [the_time]
 7  -rw-     639    Oct 02 1997 12:37:13 the_time

20578304 bytes total (3104544 bytes free)

Router# show slot0:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time----- name
1  .. unknown 317FBA1B 4A0694 24 4720148 Aug 29 1997 17:49:36 hampton/nitz
2  .. unknown 9237F3FF 92C574 11 4767328 Oct 01 1997 18:42:53 c7200-js-mz
3  .D unknown 71AB01F1 10C94E0 10 7982828 Oct 01 1997 18:48:14 rsp-jsv-mz
4  .D unknown 96DACD45 10C97E0 8 639 Oct 02 1997 12:09:17 the_time
5  .. unknown 96DACD45 10C9AE0 3 639 Oct 02 1997 12:09:32 foo
6  .D unknown 96DACD45 10C9DE0 8 639 Oct 02 1997 12:37:01 the_time
7  .. unknown 96DACD45 10CA0E0 8 639 Oct 02 1997 12:37:13 the_time

3104544 bytes available (17473760 bytes used)
```

Displaying a File

To display the contents of any readable file, including a file on a remote file system, use the following EXEC mode command:

Command	Purpose
<code>more [/ascii /binary /ebcdic] file-url</code>	Display a file.

The following example displays the contents of a configuration file on a TFTP server:

```
Router# more tftp://serverA/hampton/savedconfig
!
! Saved configuration on server
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
...
end
```

Flash Memory File System Types

Cisco platforms use one of three different Flash memory file system types:

- Class A Flash File Systems
- Class B Flash File Systems
- Class C Flash File Systems

The methods used for erasing, deleting, and recovering files depend on the class of the Flash file system. Some commands are supported on only one or two file system types. The command reference documentation notes commands that are not supported on all file system types.

Refer to Table 12 to determine which Flash memory file system type your platform uses.

Table 12 Flash Memory File System Types

Type	Platforms
Class A	Cisco 7000 family, C12000, LS1010
Class B	Cisco 1003, Cisco 1004, Cisco 1005, Cisco 2500 series, Cisco 3600 series, Cisco 4000 series, Cisco AS5200
Class C	Cisco MC3810, disk0 of SC3640

Class A Flash File Systems

On Class A Flash file systems, you can delete individual files using the **delete** command and later recover these files with the **undelete** command. The delete command marks the files as “deleted,” but the files still take up space in Flash memory. To permanently delete the files, use the **squeeze** command. The **squeeze** command removes all of the files marked “deleted” from the specified Flash memory device. These files can no longer be recovered. To erase all of the files on a Flash device, use the **format** command.

Deleting Files on a Flash Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a “deleted” file, as discussed in the following section. You may want to recover a “deleted” image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash device, use the following EXEC mode command:

Command	Purpose
<code>delete [device:] filename</code>	Delete a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the `cd` command.

If you attempt to delete the file specified by the `CONFIG_FILE` or `BOOTLDR` environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the `BOOT` environment variable, the system prompts you to confirm the deletion.

The following example deletes the file `myconfig` from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following commands in EXEC mode:

	Command	Purpose
Step 1	<code>dir /all [filesystem:]</code>	Determine the index of the deleted file.
Step 2	<code>undelete index [filesystem:]</code>	Undelete a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the “deleted” list could contain multiple configuration files with the name `router-config`. You undelete by index to indicate which of the many `router-config` files from the list to undelete. Use the `dir` command with the `/all` option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid (undeleted) one with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had an undeleted version of the `router-config` file and you wanted to use a previous, deleted version instead, you cannot simply undelete the previous version by index. You must first delete the existing `router-config` file and then undelete the previous `router-config` file by index. You can undelete a file as long as the file has not been permanently erased with the `squeeze` command. You can delete and undelete a file up to 15 times.

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Permanently Deleting Files on a Flash Device

When a Flash memory device is full, you may need to rearrange the files so that the space used by the “deleted” files can be reclaimed. To determine whether a Flash memory device is full, use the **dir** command.

To permanently delete files on a Flash memory device, use the following command from privileged EXEC mode:

Command	Purpose
<code>squeeze filesystem:</code>	Permanently delete all files marked “deleted” in Flash memory.

When you issue the **squeeze** command, the router copies all valid files to the beginning of Flash memory and erases all files marked “deleted.” At this point, you cannot recover “deleted” files, and you can now write to the reclaimed Flash memory space.



Note

The squeeze operation can take as long as several minutes because it can involve erasing and rewriting almost an entire Flash memory space.

Verifying Flash

To recompute and verify the checksum of a file in Flash memory on a Class A Flash file system, use the **verify** command.

Example: Deleting and Recovering a Class A Flash File System

In the following example, the image c7200-js-mz is deleted and undeleted. Note that the deleted file does not appear in the output for the first **dir** command, but it appears in the output for the **dir /all** command.

```
Router# delete slot1:
Delete filename []? c7200-js-mz
Delete slot1:c7200-js-mz? [confirm]
Router# dir slot1:
Directory of slot1:/

No such file

20578304 bytes total (15754684 bytes free)
Router# dir /all slot1:
Directory of slot1:/

  1  -rw-      4823492   Dec 17 1997 13:21:53  [c7200-js-mz]

20578304 bytes total (15754684 bytes free)
Router# undelete 1 slot1:
Router# dir slot1:
Directory of slot1:/

  1  -rw-      4823492   Dec 17 1997 13:21:53  c7200-js-mz

20578304 bytes total (15754684 bytes free)
```

In the following example, the image is deleted. In order to reclaim the space taken up by the deleted file, the **squeeze** command is issued.

```
Router# delete slot1:c7200-js-mz
Delete filename [c7200-js-mz]?
Delete slot1:c7200-js-mz? [confirm]
Router# squeeze slot1:
All deleted files will be removed. Continue? [confirm]
Squeeze operation may take a while. Continue? [confirm]
Erasing squeeze log
Squeeze of slot1: complete
Router# dir /all slot1:
Directory of slot1:/

No such file

20578304 bytes total (20578304 bytes free)
```

Class B Flash File Systems

On Class B Flash file systems, you can delete individual files with the **delete** command. The **delete** command marks the file as “deleted.” The file is still present in Flash memory and takes up space. To recover the file, use the **undelete** command. To reclaim any space in Flash memory, you must erase the entire Flash file system with the **erase** command.

Deleting Files on a Flash Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a “deleted” file, as discussed in the following section. You may want to recover a “deleted” image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash device, use the following EXEC mode command:

Command	Purpose
<code>delete [device:]filename</code>	Delete a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the **cd** command.

The following example deletes the file myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following EXEC mode commands:

	Command	Purpose
Step 1	<code>dir /all [filesystem:]</code>	Determine the index of the deleted file.
Step 2	<code>undelete index [filesystem:]</code>	Undelete a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the “deleted” list could contain multiple configuration files with the name `router-config`. You undelete by index to indicate which of the many `router-config` files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid (undeleted) one with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had an undeleted version of the `router-config` file and you wanted to use a previous, deleted version instead, you cannot simply undelete the previous version by index. You must first delete the existing `router-config` file and then undelete the previous `router-config` file by index. You can undelete a file as long as the file system has not been permanently erased with the **erase** command. You can delete and undelete a file up to 15 times.

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Erasing Flash Memory

In order to reclaim any space taken up by files in Flash memory, you must erase the entire file system using the **erase flash:** or **erase bootflash:** commands. These commands reclaim all of the space in Flash memory, erasing all files, deleted or not, in the process. Once erased, these files cannot be recovered. Before erasing Flash memory, save any files you wish to keep in another location (an FTP server, for example). Copy the files back to Flash memory after you have erased the device.

To erase a Flash device, use the following EXEC mode command:

Command	Purpose
<code>erase filesystem:</code>	Erases the Flash file system.

Example: Erasing a File System

The following example erases all files in the second partition in Flash memory:

```
Router# erase flash:2

System flash directory, partition 2:
File Length Name/status
  1 1711088 dirt/gate/c1600-i-mz
[1711152 bytes used, 15066064 available, 16777216 total]

Erase flash device, partition 2? [confirm]
Are you sure? [yes/no]: yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
```

Verifying Flash

To recompute and verify the checksum of a file in Flash memory on a Class B Flash file system, use the **verify** command.

Class C Flash File Systems

On Class C Flash memory file systems, you can delete individual files with the **delete** command. Files cannot be reclaimed once they have been deleted. Instead, the Flash file system space is reclaimed dynamically. To erase all of the files in Flash, use the **format** command.

Deleting Files on a Flash Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file on a Class C file system, the file is deleted permanently. The router reclaims the space dynamically.

To delete a file from a specified Flash device, use the following EXEC mode command:

Command	Purpose
<code>delete [device:] filename</code>	Deletes a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the **cd** command.

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

The following example permanently deletes the *myconfig* file from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Formatting Flash

To format a Class C Flash file system, use the following EXEC mode command:

Command	Purpose
<code>format filesystem</code>	Formats Flash.

If you format a Flash device, all of the files are erased and cannot be recovered.

Creating and Removing Directories

On Class C Flash file systems, you can create a new directory with the **mkdir** command. To remove a directory from a Flash file system, use the **rmdir** command.

On Class C Flash file systems, you can rename a file using the **rename** command.

Checking Flash File Systems

On Class C Flash file systems, you can check a file system for damage and repair any problems using the **fsck** command.

Remote File System Management

On remote file systems (file systems on FTP, rcp, or TFTP servers) you can perform the following tasks:

- View the contents of a file with the **more** command.
- Copy files to or from the router using the **copy** command.
- Show information about a file using the **show file information** command.



Note

You cannot delete files on remote systems.

NVRAM File System Management

On most platforms, NVRAM contains the startup configuration. On Class A Flash file system platforms, the CONFIG_FILE environment variable specifies the location of the startup configuration. However, the file URL `nvrाम:startup-config` always specifies the startup configuration, regardless of the CONFIG_FILE environment variable.

You can view the startup-config (with the **more nvrाम:startup-config** command), replace the startup config with a new configuration file (with the **copy source-url nvrाम:startup-config** command), save the startup configuration to another location (with the **copy nvrाम:startup-config destination-url** command), and erase the contents of NVRAM (with the **erase nvrाम:** command). The **erase nvrाम:** command also deletes the startup configuration if another location is specified by the CONFIG_FILE variable.

This example displays the startup configuration:

```

nnm3640-2# more nvrाम:startup-config
Using 2279 out of 129016 bytes
!
! Last configuration change at 10:57:25 PST Wed Apr 22 1998
! NVRAM config last updated at 10:57:27 PST Wed Apr 22 1998
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
...
end

```

The following example displays the contents of the NVRAM file system on a Class A Flash file system platform. The file named `startup-config` is the current startup configuration file, in physical NVRAM or in Flash memory. If the file is located in a Flash memory file system, this entry is a symbolic link to the actual file. The file named `underlying-config` is always the NVRAM version of the configuration.

```

Router# dir nvrाम:
Directory of nvrाम:/

   1  -rw-          2703          <no date>  startup-config
   2  ----           5          <no date>  private-config
   3  -rw-          2703          <no date>  underlying-config

129016 bytes total (126313 bytes free)

```

System File System Management

The system file system contains the system memory and the current running configuration. You can view the current configuration (with the **show running-config** or **more system:running-config** command), save the current configuration to some other location (with the **copy system:running-config destination-url** command), or add configuration commands to the current configuration (with the **copy source-url system:running-config** command).

The following example changes to the system file system, views the contents of the file system, and displays the running configuration:

```

nnm3640-2# cd system:
nnm3640-2# dir
Directory of system:/

   2  dr-x          0          <no date>  memory
   1  -rw-          0          <no date>  running-config

No space information available
nnm3640-2# show running-config
!
! No configuration change since last restart
!
version 11.3
service timestamps log datetime localtime
service linenumbers
service udp-small-servers
service pt-vty-logging
!
...
end

```

On some platforms, the system file system contains microcode in its ucode directory.

```

Router# dir system:/ucode
Directory of system:/ucode/

   21  -r--          22900          <no date>  aip20-13
   18  -r--          32724          <no date>  eip20-3
   25  -r--          123130         <no date>  feip20-6
   19  -r--          25610          <no date>  fip20-1
   22  -r--           7742          <no date>  fsip20-7
   23  -r--          17130          <no date>  hip20-1
   24  -r--          36450          <no date>  mip22-2
   29  -r--          154752         <no date>  posip20-0
   28  -r--          704688         <no date>  rsp220-0
   20  -r--          33529          <no date>  trip20-1
   26  -r--          939130         <no date>  vip22-20
   27  -r--          1107862        <no date>  vip222-20

No space information available

```

