



Managing Configuration Files

Last Updated: May 4, 2009

This chapter describes how to create, load, and maintain configuration files. Configuration files contain a set of user-configured commands that customize the functionality of your Cisco routing device.

The tasks in this chapter assume that you have at least a minimal configuration running on your system. You can create a basic configuration file using the **setup** command (see [Using Setup Mode to Configure a Cisco Networking Device](#) for details).

For a complete description of the configuration file management commands in this chapter, refer to the [Cisco IOS Configuration Fundamentals Command Reference](#).

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see [About Cisco IOS Software Documentation](#).

Finding Feature Information

For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Managing Configuration Files”](#) section on page 19.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Understanding Configuration Files, page 2](#)
- [Configuration File Management Task List, page 3](#)
- [Displaying Configuration File Information, page 3](#)
- [Entering Configuration Mode and Selecting a Configuration Source, page 4](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2003-2008 Cisco Systems, Inc. All rights reserved.

- [Modifying the Configuration File at the CLI](#), page 4
- [Copying Configuration Files from the Router to a Network Server](#), page 5
- [Copying Configuration Files from a Network Server to the Router](#), page 6
- [Maintaining Configuration Files Larger than NVRAM](#), page 7
- [Controlling the Parser Cache](#), page 10
- [Copying Configuration Files Between Different Locations](#), page 12
- [Reexecuting the Configuration Commands in the Startup Configuration File](#), page 14
- [Clearing Configuration Information](#), page 14
- [Specifying the Startup Configuration File](#), page 15
- [Technical Assistance](#), page 18
- [Feature Information for Managing Configuration Files](#), page 19

Understanding Configuration Files

Configuration files contain the Cisco IOS XE software commands used to customize the functionality of your Cisco routing device (router, access server, switch, and so on). Commands are parsed (translated and executed) by the Cisco IOS XE software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

Types of Configuration Files

Startup configuration files (startup-config) are used during system startup to configure the software. Running configuration files (running-config) contain the current configuration of the software. The two configuration files can be different. For example, you may want to change the configuration for a short time period rather than permanently. In this case, you would change the running configuration using the **configure terminal EXEC** command but not save the configuration using the **copy running-config startup-config EXEC** command.

To change the running configuration, use the **configure terminal** command, as described in the “[Modifying the Configuration File at the CLI](#)” section later in this chapter. As you use the Cisco IOS XE configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup configuration file, you can either save the running configuration file to the startup configuration using the **copy running-config startup-config EXEC** command or copy a configuration file from a file server to the startup configuration (see the “[Copying Configuration Files from a Network Server to the Router](#)” section for more information).

Location of Configuration Files

Configuration files are stored in the following locations:

- The running configuration is stored in RAM.
- On all platforms except the Class A Flash file system platforms, the startup configuration is stored in nonvolatile random-access memory (NVRAM).

- On Class A Flash file system platforms, the startup configuration is stored in the location specified by the CONFIG_FILE environment variable (see the “[Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems](#)” section for more information). The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
 - **nvr**am: (NVRAM)
 - **bootflash**: (internal Flash memory)
 - **HDD**: (harddisk)
 - **usb0**: (external USB media 1)
 - **usb1**: (external USB media 2)

Configuration File Management Task List

To understand the management of Cisco IOS XE software configuration files, perform the tasks described in the following sections:

- [Displaying Configuration File Information, page 3](#)
- [Entering Configuration Mode and Selecting a Configuration Source, page 4](#)
- [Modifying the Configuration File at the CLI, page 4](#)
- [Copying Configuration Files from the Router to a Network Server, page 5](#)
- [Copying Configuration Files from a Network Server to the Router, page 6](#)
- [Maintaining Configuration Files Larger than NVRAM, page 7](#)
- [Controlling the Parser Cache, page 10](#)
- [Copying Configuration Files Between Different Locations, page 12](#)
- [Reexecuting the Configuration Commands in the Startup Configuration File, page 14](#)
- [Clearing Configuration Information, page 14](#)
- [Specifying the Startup Configuration File, page 15](#)

Displaying Configuration File Information

To display information about configuration files, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# 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.
Router# more file-url	Displays the contents of a specified file.

Command	Purpose
Router# show running-config	Displays the contents of the running configuration file. (Command alias for the more system:running-config command.)
Router# show startup-config	Displays the contents of the startup configuration file. (Command alias for the more nvram:startup-config command.) On all platforms except the Class A Flash file system platforms, the default startup-config file usually is stored in NVRAM. On the Class A Flash file system platforms, the CONFIG_FILE environment variable points to the default startup-config file. The CONFIG_FILE variable defaults to NVRAM.

Entering Configuration Mode and Selecting a Configuration Source

To enter configuration mode on the router, enter the **configure** command at the privileged EXEC prompt. The Cisco IOS XE software responds with the following prompt asking you to specify the terminal, memory, or a file stored on a network server (network) as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

Configuring from the terminal allows you to enter configuration commands at the command line, as described in the following section. Configuring from memory loads the startup configuration file. See the “[Reexecuting the Configuration Commands in the Startup Configuration File](#)” section for more information. Configuring from the network allows you to load and execute configuration commands over the network. See the “[Copying Configuration Files from a Network Server to the Router](#)” section for more information.

Modifying the Configuration File at the CLI

The Cisco IOS XE software accepts one configuration command per line. You can enter as many configuration commands as you want.

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the **show running-config** or **more system:running-config** EXEC command. Comments do not display when you list the startup configuration with the **show startup-config** or **more nvram:startup-config** EXEC mode command. Comments are stripped out of the configuration file when it is loaded onto the router. However, you can list the comments in configuration files stored on a Trivial File Transfer Protocol (TFTP) server.

When you configure the software using the CLI, the software executes the commands as you enter them. To configure the software using the CLI, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2		Enter the necessary configuration commands. The Cisco IOS XE documentation set describes configuration commands organized by technology.
Step 3	Router(config)# end or Router(config)# ^Z	Ends the configuration session and exits to EXEC mode. Note When you press the Ctrl and Z keys simultaneously, ^Z is displayed to the screen.
Step 4	Router# copy system:running-config nvram:startup-config	Saves the running configuration file as the startup configuration file. You may also use the copy running-config startup-config command alias, but you should be aware that this command is less precise. On most platforms, this command saves the configuration to NVRAM. On the Class A Flash file system platforms, this step saves the configuration to the location specified by the CONFIG_FILE environment variable (the default CONFIG_FILE variable specifies that the file should be saved to NVRAM).

In the following example, the router prompt name of the router is configured. The comment line, indicated by the exclamation mark (!), does not execute any command.

In this example, the **hostname** command is used to change the router name from Router to new_name. By pressing Ctrl-Z (^Z) or entering the **end** command, the user quits configuration mode. The **copy system:running-config nvram:startup-config** command saves the current configuration to the startup configuration.

```
Router# configure terminal
Router(config)# !The following command provides the router host name.
Router(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

When the startup configuration is NVRAM, it stores the current configuration information in text format as configuration commands, recording only nondefault settings. The memory is checksummed to guard against corrupted data.


Note

Some specific commands might not get saved to NVRAM. You will need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a list of these settings so that you can quickly reconfigure your router after rebooting.

Copying Configuration Files from the Router to a Network Server

You can copy configuration files from the router to a file server using TFTP. For example, you might perform this task to back up a current configuration file to a server before changing its contents, thereby allowing you to later restore the original configuration file from the server.

To copy configuration files from a router to a server, perform the tasks described in the following sections:

- [Copying a Configuration File from the Router to a TFTP Server](#)

Copying a Configuration File from the Router to a TFTP Server

In some implementations of TFTP, you must create a dummy file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

To copy configuration information on a TFTP network server, use the following commands in the EXEC mode, as needed:

Command	Purpose
Router# copy system:running-config tftp:[[//location]/directory]/filename]	Copies the running configuration file to a TFTP server.
Router# copy nvram:startup-config tftp:[[//location]/directory]/filename]	Copies the startup configuration file to a TFTP server.

After you have issued the **copy** command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

The following example copies a configuration file from a router to a TFTP server:

```
Tokyo# copy system:running-config tftp://172.16.2.155/tokyo-config
Write file tokyo-config on host 172.16.2.155? [confirm] y

Writing tokyo-config!!! [OK]
```

Copying Configuration Files from a Network Server to the Router

You can copy configuration files from a TFTP server to the running configuration or startup configuration of the router. You may want to perform this function for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another router. For example, you may add another router to your network and want it to have a similar configuration to the original router. By copying the file to the new router, you can change the relevant parts rather than re-creating the whole file.
- To load the same configuration commands on to all the routers in your network so that all the routers have similar configurations.

The **copy tftp: system:running-config** EXEC command loads the configuration files into the router as if you were typing the commands in at the command line. The router does not erase the existing running configuration before adding the commands. If a command in the copied configuration file replaces a command in the existing configuration file, the existing command will be erased. For example, if the copied configuration file contains a different IP address in a particular command than the existing configuration, the IP address in the copied configuration will be used. However, some commands in the existing configuration may not be replaced or negated. In this case, the resulting configuration file will be a mixture of the existing configuration file and the copied configuration file, with the copied configuration file having precedence.

In order to restore a configuration file to an exact copy of a file stored on a server, you need to copy the configuration file directly to the startup configuration (using the **copy tftp: nvram:startup-config** command) and reload the router.

To copy configuration files from a server to a router, perform the tasks described in the following sections:

- [Copying a Configuration File from a TFTP Server to the Router](#)

Copying a Configuration File from a TFTP Server to the Router

To copy a configuration file from a TFTP server to the router, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# copy tftp: [[[//location]/directory]/filename] system:running-config	Copies a configuration file from a TFTP server to the running configuration.
Router# copy tftp: [[[//location]/directory]/filename] nvrām:startup-config	Copies a configuration file from a TFTP server to the startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

In the following example, the software is configured from the file named `tokyo-config` at IP address 172.16.2.155:

```
Router1# copy tftp://172.16.2.155/tokyo-config system:running-config
Configure using tokyo-config from 172.16.2.155? [confirm] y
Booting tokyo-config from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

Maintaining Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds size of NVRAM, perform the tasks described in the following sections:

- [Compressing the Configuration File](#)
- [Storing the Configuration in Flash Memory on Class A Flash File Systems](#)
- [Loading the Configuration Commands from the Network](#)

Compressing the Configuration File

The **service compress-config** global configuration command specifies that the configuration file be stored compressed in NVRAM. Once the configuration file has been compressed, the router functions normally. When the system is booted, it recognizes that the configuration file is compressed, expands it, and proceeds normally. The **more nvrām:startup-config** EXEC command expands the configuration before displaying it.

Before you compress configuration files, refer to the appropriate hardware installation and maintenance publication. Verify that your system's ROMs support file compression. If not, you can install new ROMs that support file compression.

To compress configuration files, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# service compress-config	Specifies that the configuration file be compressed.
Step 2	Router(config)# end	Exits global configuration mode.
Step 3	Use TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - file-size/buffer-size bytes]." or Router# configure terminal	Enters the new configuration.
Step 4	Router(config)# copy system:running-config nvram:startup-config	When you have finished changing the running-configuration, saves the new configuration.

The size of the configuration must not exceed three times the NVRAM size. For a 128-KB size NVRAM, the largest expanded configuration file size is 384 KB.

The **service compress-config** global configuration command works only if you have Cisco IOS XE software Release 10 or later release boot ROMs. If the boot ROMs do not recognize a compressed configuration, the following message is displayed:

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

The following example compresses a 129-KB configuration file to 11 KB:

```
Router# configure terminal
Router(config)# service compress-config
Router(config)# end
Router# copy tftp://172.16.2.15/tokyo-config system:running-config
Configure using tokyo-config from 172.16.2.155? [confirm] y
Booting tokyo-config from 172.16.2.155:!!! [OK - 874/16000 bytes]
Router# copy system:running-config nvram:startup-config
Building configuration...
Compressed configuration from 2654 bytes to 1332 bytes[OK]
Uncompressed configuration from 1332 bytes to 2654 bytes
```

Storing the Configuration in Flash Memory on Class A Flash File Systems

On Class A Flash file system routers, you can store the startup configuration in Flash memory by setting the CONFIG_FILE environment variable to a file in internal Flash memory or Flash memory in a USB port.

To store the startup configuration in Flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# copy nvram:startup-config flash-filesystem:filename	Copies the current startup configuration to the new location to create the configuration file.
Step 2	Router# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 3	Router(config)# boot config filesystem: <i>filename</i>	Specifies that the startup configuration file be stored in Flash memory by setting the CONFIG_FILE variable.
Step 4	Router(config)# end	Exits global configuration mode.
Step 5	Use TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - file-size/buffer-size bytes]." or Router# configure terminal	Enters the new configuration.
Step 6	Router# copy system:running-config nvram:startup-config	When you have finished changing the running-configuration, saves the new configuration.

See the “[Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems](#)” section for more information.

The following example stores the configuration file in usb 0:

```
Router# copy nvram:startup-config usb0:router-config
Router# configure terminal
Router(config)# boot config usb0:router-config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Care must be taken when editing or changing a large configuration. Flash memory space is used every time a **copy system:running-config nvram:startup-config EXEC** command is issued. Because file management for Flash memory, such as optimizing free space, is not done automatically, you must pay close attention to available Flash memory. Use the **squeeze** command to reclaim used space. We recommend that you use a large-capacity Flash card of at least 20 MB.

Loading the Configuration Commands from the Network

You can also store large configurations on TFTP servers and download them at system startup. To use a network server to store large configurations, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# copy system:running-config {tftp:}	Saves the running configuration to an TFTP server.
Step 2	Router# configure terminal	Enters global configuration mode.
Step 3	Router(config)# boot network { tftp: [[<i>//location</i>]/ <i>directory</i>]/ <i>filename</i>]	Specifies that the startup configuration file be loaded from the network server at startup.
Step 4	Router(config)# service config	Enables the router to download configuration files at system startup.
Step 5	Router(config)# end	Exits global configuration mode.
Step 6	Router# copy system:running-config nvram:startup-config	Saves the configuration.

See the “[Copying Configuration Files from the Router to a Network Server](#)” and “[Configuring the Router to Download Configuration Files](#)” sections for more information on these commands.

Controlling the Parser Cache

The Cisco IOS XE command-line parser in the Cisco IOS XE software performs the translation and execution (parsing) of command lines. The Parser Cache feature was developed to rapidly process large configuration files, thereby dramatically improving load time.

The Parser Cache feature allows the rapid recognition and translation of configuration lines in a configuration file that differ slightly from previously used configuration lines (for example, `pvc 0/100`, `pvc 0/101`, and so on) by dynamically creating, caching, and reusing simplified parse graphs. This improvement is useful primarily for configuration files that repeat similar commands hundreds or thousands of times, such as cases in which thousands of virtual circuits must be configured for subinterfaces, or hundreds of access lists must be configured. Performance will improve the most for those files in which the same commands are used repeatedly but the numerical arguments change from command to command.

The Parser Cache is enabled by default. However, users with Cisco devices that do not require large configuration files may want to disable the Parser Cache to free the resources used by this feature. (Memory used by this feature depends on the size of the configuration files parsed, but is generally less than 512 KB.)

To control the Parser Cache feature, perform the tasks described in the following sections. All of these tasks are optional:

- [Clearing the Parser Cache](#)
- [Disabling the Parser Cache](#)
- [Reenabling the Parser Cache](#)
- [Monitoring the Parser](#)

Clearing the Parser Cache

To free resources or to reset the parser cache memory, you may wish to clear the parse entries and hit/miss statistics stored by the Parser Cache feature. To clear the information stored by the Parser Cache feature, use the following command in privileged EXEC mode:

Command	Purpose
Router# <code>clear parser cache</code>	Clears the parse cache entries and hit/miss statistics stored for the Parser Cache feature.

Disabling the Parser Cache

The Parser Cache feature is enabled by default. To disable the Parser Cache feature, use the following command in global configuration mode:

Command	Purpose
Router(config)# no parser cache	Disables the Parser Cache feature.

When the parser cache is disabled, the **no parser cache** command line is written to the running configuration file.



Tip

If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **no parser cache** command. You will not be able to clear the parser cache after disabling it.

Reenabling the Parser Cache

To reenable the Parser Cache feature after disabling it, use the following command in global configuration mode:

Command	Purpose
Router(config)# parser cache	Enables the Parser Cache feature.

Monitoring the Parser

Statistics about the last configuration file parsed are kept in the system memory, along with hit/miss statistics on the commands parsed by the Parser Cache feature. “Hits” and “misses” refer to the matches that the parser cache was able to make to similar commands used previously in the configuration session. Those commands that are matched (“hits”) be parsed more efficiently. The parser cache cannot improve the parse time for those commands it was unable to match (“misses”).

To display the parser statistics, use the following command in privileged EXEC mode:

Command	Purpose
Router# show parser statistics	Displays statistics about the last configuration file parsed and the status of the Parser Cache feature.

The following example shows sample output from the **show parser statistics** command:

```
Router# show parser statistics
Last configuration file parsed: Number of Commands:1484, Time:1272 ms

Parser cache:disabled, 0 hits, 0 misses
```

The **show parser statistics** command displays two sets of data, as follows:

- The number of commands in the configuration file that was last copied into the running configuration, and the time it took for the system to parse them (a configuration file can be loaded into the running configuration at system startup, or by issuing commands such as the **copy source running-config EXEC** command).
- The status of the parser cache (enabled or disabled) and the number of command matches (hits or misses) since the system was started or since the parser cache was cleared.

In the example shown, the hit/miss statistics (0/0) do not match the number of commands in the last configuration file parsed (1484), which indicates that the last configuration file was loaded while the parser cache was disabled.

Copying Configuration Files Between Different Locations

On many platforms, you can copy configuration files from one Flash memory device, such as internal Flash memory or a Flash memory attached to a USB port, to other locations. You also can copy configuration files from an TFTP server to Flash memory.

Copying Configuration Files from Flash Memory to the Startup or Running Configuration

To copy a configuration file from Flash memory directly to your startup configuration in NVRAM or your running configuration, enter one following commands in EXEC mode, as needed:

Command	Purpose
Router> copy filesystem:[partition-number:] [filename] nvram:startup-config	Loads a configuration file directly into NVRAM.
Router> copy filesystem:[partition-number:] [filename] system:running-config	Copies a configuration file to your running configuration.

The following example copies the file named ios-upgrade-1 from partition 4 of the Flash memory PC Card in usb 0 to the router startup configurations:

```
Router# copy bootflash: nvram:startup-config
Source filename []? 50K_ACL-config
Destination filename [startup-config]?
Compressed configuration from 2580593 bytes to 207846 bytes[OK]
2580593 bytes copied in 39.059 secs (66069 bytes/sec)
```

Copying Configuration Files Between Flash Memory File Systems

On platforms with multiple Flash memory file systems, you can copy files from one Flash memory file system, such as internal Flash memory or a Flash memory card, to another Flash memory file system. Copying files to different Flash memory file systems lets you create backup copies of working configurations and duplicate configurations for other routers.

To copy a configuration file between Flash memory file systems, use the following commands in EXEC mode:

	Command	Purpose
Step 1	Router> show source-filesystem:	Displays the layout and contents of Flash memory to verify the filename.
Step 2	Router> copy source-filesystem: [partition-number:] [filename] dest-filesystem: [partition-number:] [filename]	Copies a configuration file between Flash memory devices.
Step 3	Router> verify dest-filesystem: [partition-number:] [filename]	Verifies the checksum of the file you copied.

Copying a Configuration File Between Local Flash Memory Devices Example

The following example copies the file named running-config from partition 1 of internal Flash memory to partition 1 of usb 1 on a ASR1000 series router. In this example, the source partition is not specified, so the router prompts for the partition number.

```
Router# copy bootflash: usb0:
Source filename [50K_ACL-config]?
Destination filename [50K_ACL-config]?
Copy in progress...CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2580593 bytes copied in 0.473 secs (5455799 bytes/sec)

Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)... OK

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy 'running-config' from flash: device
as 'running-config' into slot1: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
...erased
!
[OK - 850/4194304 bytes]

Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum... OK (0x16)
```

Copying a Configuration File from a Server to Flash Memory Devices

To copy a configuration file from a TFTP server to the router, use the following command in EXEC mode:

Command	Purpose
Router> copy tftp: [[[//location]/directory]/filename] flash-filesystem: [partition-number:] [filename]	Copies the file from a TFTP server to the Flash memory device. Reply to any router prompts for additional information or confirmation. The prompting will depending on how much information you provide in the copy command and the current setting of the file prompt command.

The following example shows the copying of the configuration file named `router-config` from a TFTP server to the Flash memory inserted in `usb 0` of a Cisco ASR1000 series router. The copied file is renamed `new-config`.

```
Router# copy tftp:router-config usb0:new-config
```

Reexecuting the Configuration Commands in the Startup Configuration File

To reexecute the commands located in the startup configuration file, use the following command in privileged EXEC mode:

Command	Purpose
Router# <code>configure memory</code>	Reexecutes the configuration commands located in the startup configuration file.

Clearing Configuration Information

You can clear the configuration information from the startup configuration. If you reboot the router with no startup configuration, the router will enter the Setup command facility so that you can configure the router from scratch.

Clearing the Startup Configuration

To clear the contents of your startup configuration, use the following command in EXEC mode:

Command	Purpose
Router> <code>erase nvram:</code>	Clears the contents of your startup configuration.

For all platforms except the Class A Flash file system platforms, this command erases NVRAM. The startup configuration file cannot be restored once it has been deleted.

On Class A Flash file system platforms, when you use the `erase startup-config` EXEC command, the router erases or deletes the configuration pointed to by `CONFIG_FILE` environment variable. If this variable points to NVRAM, the router erases NVRAM. If the `CONFIG_FILE` environment variable specifies a Flash memory device and configuration filename, the router deletes the configuration file. That is, the router marks the file as “deleted,” rather than erasing it. This feature allows you to recover a deleted file.

Deleting a Specified Configuration File

To delete a specified configuration on a specific Flash device, use the following command in EXEC mode:

Command	Purpose
Router> delete <i>flash-filesystem:filename</i>	Deletes a specified configuration file on a specified Flash device.

On Class A and B Flash file systems, when you delete a specific file in Flash memory, the system marks the file as deleted. Deleted files cannot be recovered.

If you attempt to delete the configuration file specified by the CONFIG_FILE environment variable, the system prompts you to confirm the deletion.

The following example deletes the file named myconfig from a Flash memory inserted in usb 0:

```
Router# delete usb0:myconfig
```

Specifying the Startup Configuration File

Normally, the router uses the startup configuration file in NVRAM or the Flash file system specified by the CONFIG_FILE environment variable (Class A Flash file systems only) at startup. See the “[Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems](#)” section for more information on setting the CONFIG_FILE variable.

You can also configure the router to automatically request and receive two configuration files from the network server at startup. See the “[Configuring the Router to Download Configuration Files](#)” section for more information.

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems

On Class A Flash file systems, you can configure the Cisco IOS XE software to load the startup configuration file specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM. To change the CONFIG_FILE environment variable, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	Router> copy [<i>flash-url</i> <i>tftp-url</i> system:running-config nvrasm:startup-config] <i>dest-flash-url</i>	Copies the configuration file to the Flash file system from which the router will load the file upon restart.
Step 2	Router# configure terminal	Enters global configuration mode.
Step 3	Router(config)# boot config <i>dest-flash-url</i>	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
Step 4	Router(config)# end	Exits global configuration mode.
Step 5	Router> copy system:running-config nvrasm:startup-config	Saves the configuration performed in Step 3 to the startup configuration.
Step 6	Router> show bootvar	(Optional) Allows you to verify the contents of the CONFIG_FILE environment variable.

After you specify a location for the startup configuration file, the **nvr_{am}:startup-config** command is aliased to the new location of the startup configuration file. The **more nvr_{am}:startup-config EXEC** command will display the startup configuration, regardless of its location. The **erase nvr_{am}:startup-config EXEC** command will erase the contents of NVRAM and delete the file pointed to by the CONFIG_FILE environment variable.

When you save the configuration using the **copy system:running-config nvr_{am}:startup-config** command, the router saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and a distilled version to NVRAM. A distilled version is one that does not contain access list information. If NVRAM contains a complete configuration file, the router prompts you to confirm your overwrite of the complete version with the distilled version. If NVRAM contains a distilled configuration, the router does not prompt you for confirmation and proceeds with overwriting the existing distilled configuration file in NVRAM.



Note

If you specify a file in a Flash device as the CONFIG_FILE environment variable, every time you save your configuration file with the **copy system:running-config nvr_{am}:startup-config** command, the old configuration file is marked as “deleted,” and the new configuration file is saved to that device. Eventually, Flash memory will be full, because the old configuration files still take up memory. Use the **squeeze EXEC** command to permanently delete the old configuration files and reclaim the space.

The following example copies the running configuration file to the first USB port of a Cisco ASR1000 series router. This configuration is then used as the startup configuration when the system is restarted.

```
Router# copy system:running-config usb0:config2
Router# configure terminal
Router(config)# boot config usb0:config2
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
Router# show bootvar
BOOT variable = usb0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = usb0:config2

Configuration register is 0x010F
```

Configuring the Router to Download Configuration Files

You can configure the router to load one or two configuration files at system startup. The configuration files are loaded into memory and read in as if you were typing the commands at the command line. Thus, the configuration for the router will be a mixture of the original startup configuration and the one or two downloaded configuration files.

Network Versus Host Configuration Files

For historical reasons, the first file the router downloads is called the network configuration file. The second file the router downloads is called the host configuration file. Two configuration files can be used when all of the routers on a network use many of the same commands. The network configuration file contains the standard commands used to configure all of the routers. The host configuration files contain the commands specific to one particular host. If you are loading two configuration files, the host configuration file should be the configuration file you want to have precedence over the other file. Both the network and host configuration files must reside on a network server reachable via TFTP.

Configuring the Router to Download Configuration Files

You can specify an ordered list of network configuration and host configuration filenames. The Cisco IOS XE software scans this list until it loads the appropriate network or host configuration file.

To configure the router to download configuration files at system startup, perform at least one of the tasks described in the following sections:

- [Configuring the Router to Download the Network Configuration File](#)
- [Configuring the Router to Download the Host Configuration File](#)

If the router fails to load a configuration file during startup, it tries again every 10 minutes (the default setting) until a host provides the requested files. With each failed attempt, the router displays the following message on the console terminal:

```
Booting host-config... [timed out]
```

Refer to the *Internetwork Troubleshooting Guide* for troubleshooting procedures.

If there are any problems with the startup configuration file, or if the configuration register is set to ignore NVRAM, the router enters the Setup command facility. See the “Using the Setup Command Facility for Configuration Changes” chapter in this publication for details on the Setup command facility.

Configuring the Router to Download the Network Configuration File

To configure the Cisco IOS XE software to download a network configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# boot network { tftp: [[<i>//location</i>]/ <i>directory</i>]/ <i>filename</i> }]	Specifies the network configuration file to download at startup, and the protocol to be used (TFTP).
Step 3	Router(config)# service config	Enables the system to automatically load the network file upon restart.
Step 4	Router(config)# end	Exits global configuration mode.
Step 5	Router# copy system:running-config nvrām:startup-config	Saves the running configuration to the startup configuration file.

For Step 2, if you do not specify a network configuration filename, the Cisco IOS XE software uses the default filename `network-config`. If you omit the address, the router uses the broadcast address.

You can specify more than one network configuration file. The software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Configuring the Router to Download the Host Configuration File

To configure the Cisco IOS XE software to download a host configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# boot host { tftp: [[[//location]/directory]/filename] }	Specifies the host configuration file to download at startup, and the protocol to be used (TFTP).
Step 3	Router(config)# service config	Enables the system to automatically load the host file upon restart.
Step 4	Router(config)# end	Exits global configuration mode.
Step 5	Router# copy system:running-config nvrnram:startup-config	Saves the running configuration to the startup configuration file.

If you do not specify a host configuration filename, the router uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending “-confg.” If no host name information is available, the software uses the default host configuration filename router-confg. If you omit the address, the router uses the broadcast address.

You can specify more than one host configuration file. The Cisco IOS XE software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Configuring the Router to Download Configuration Files at System Startup Example

In the following example, a router is configured to download the host configuration file named hostfile1 and the network configuration file named networkfile1. The router uses TFTP and the broadcast address to obtain the file.

```
Router# configure terminal
Router(config)# boot host tftp:hostfile1
Router(config)# boot network tftp:networkfile1
Router(config)# service config
Router(config)# end
Router# copy system:running-config nvrnram:startup-config
```

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password..</p>	http://www.cisco.com/techsupport

Feature Information for Managing Configuration Files

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Configuration File Management Features

Feature Name	Releases	Feature Information
Parser Cache	Cisco IOS XE Release 2.1	<p>The Cisco IOS XE command-line parser in the Cisco IOS XE software performs the translation and execution (parsing) of command lines. The Parser Cache feature was developed to rapidly process large configuration files, thereby dramatically improving load time.</p> <p>For information about feature support in Cisco IOS XE software, use Feature Navigator.</p> <p>In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> Controlling the Parser Cache

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003-2009 Cisco Systems, Inc. All rights reserved.

