



Managing Configuration Files Configuration Guide, Cisco IOS XE Everest 3.18SP (Cisco NCS 4200 Series)

First Published: 2016-07-29

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

CHAPTER 1

Managing Configuration Files 1

Finding Feature Information	1
Prerequisites for Managing Configuration Files	1
Restrictions for Managing Configuration Files	1
Information About Managing Configuration Files	2
Types of Configuration Files	2
Configuration Mode and Selecting a Configuration Source	2
Configuration File Changes Using the CLI	2
Location of Configuration Files	3
Copy Configuration Files from a Network Server to the Router	3
Copy a Configuration File from the Router to a TFTP Server	4
Copy a Configuration File from the Router to an FTP Server	4
Copying Files Through a VRF	4
Configuration Files Larger than NVRAM	5
Compressing the Configuration File	5
Storing the Configuration in Flash Memory on Class A Flash File Systems	5
Loading the Configuration Commands from the Network	5
Control of the Parser Cache	6
Configure the Router to Download Configuration Files	6
Network Versus Host Configuration Files	6
How to Manage Configuration File Information	7
Displaying Configuration File Information	7
Modifying the Configuration File at the CLI	8
Copying a Configuration File from the Router to a TFTP Server	9
What to Do Next	10
Copying a Configuration File from the Router to the FTP Server	10

Examples	12
What to Do Next	12
Copying a Configuration File from a TFTP Server to the Router	12
What to Do Next	13
Copying a Configuration File from an FTP Server to the Router	13
Examples	15
What to Do Next	15
Maintaining Configuration Files Larger than NVRAM	16
Compressing the Configuration File	16
Storing the Configuration in Flash Memory on Class A Flash File Systems	17
Loading the Configuration Commands from the Network	19
Managing the Parser Cache	20
Clearing the Parser Cache	20
Disabling the Parser Cache	21
Reenabling the Parser Cache	21
What to Do Next	22
Copying Configuration Files from Flash Memory to the Startup or Running Configuration	22
Copying a Configuration File from an FTP Server to Flash Memory Devices	23
What to Do Next	24
Copying a Configuration File from an rcp Server to Flash Memory Devices	24
Copying a Configuration File from a TFTP Server to Flash Memory Devices	25
Reexecuting the Configuration Commands in the Startup Configuration File	26
Clearing the Startup Configuration	27
Deleting a Specified Configuration File	27
Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems	28
What to Do Next	30
Configuring the Router to Download the Network Configuration File	30
Configuring the Router to Download the Host Configuration File	31
Additional References	33

CHAPTER 2
Exclusive Configuration Change Access and Access Session Locking 35

Finding Feature Information	35
Information About Locking the Configuration	36
Exclusive Configuration Change Access and Access Session Locking	36

Access Session Locking	36
How to Configure Configuration Exclusive Configuration Change-Access and Access Session Locking	37
Enabling Exclusive Configuration Change Access and Access Session Locking	37
Obtaining Exclusive Configuration Change Access	38
Monitoring and Troubleshooting Configuration Locking	39
Configuration Examples for Locking the Configuration	40
Configuring an Exclusive Lock in Auto Mode Example	40
Configuring an Exclusive Lock in Manual Mode Example	40
Additional References	41

CHAPTER 3

Configuration Partitioning	43
Finding Feature Information	43
Information About Configuration Partitioning	44
System Running Configurations	44
Retrieving the Running Configuration for Display or Copy Operations	44
Benefits of Partitioning the Running Configuration	44
How to Use the Configuration Partitioning Feature	45
Displaying Configuration Partitions	45
Disabling the Configuration Partitioning Feature	47
What to Do Next	47
Configuration Examples for Configuration Partitioning	48
Displaying Configuration Partitions Example	48
Additional References	57

CHAPTER 4

Configuration Versioning	59
Finding Feature Information	59
Information About Configuration Versioning	59
Configuration Archive	59
How to Configure Configuration Versioning	60
Configuring the Characteristics of the Configuration Archive	60
Monitoring and Troubleshooting the Configuration	62
Configuration Examples for Configuration Versioning	64
Example: Creating a Configuration Archive	64

Additional References 64

CHAPTER 5

Configuration Rollback Confirmed Change 67

Finding Feature Information 67

Information About Configuration Rollback Confirmed Change 67

Configuration Rollback Confirmed Change Operation 67

Restrictions 68

How to Configure Configuration Rollback Confirmed Change 68

Performing a Configuration Replace or Configuration Rollback Operation with Confirmation 68

Configuration Examples for Configuration Rollback Confirmed Change 70

Example: Performing a Configuration Replace Operation with the `configure confirm` Command
70

Additional References 71



CHAPTER 1

Managing Configuration Files

Creating, loading, and maintaining configuration files enable you to generate a set of user-configured commands to customize the functionality of your Cisco routing device. For a complete description of the configuration file management commands, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

- [Finding Feature Information, on page 1](#)
- [Prerequisites for Managing Configuration Files, on page 1](#)
- [Restrictions for Managing Configuration Files, on page 1](#)
- [Information About Managing Configuration Files, on page 2](#)
- [How to Manage Configuration File Information, on page 7](#)
- [Additional References, on page 33](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Prerequisites for Managing Configuration Files

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.
- You should 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).

Restrictions for Managing Configuration Files

- Many of the Cisco IOS commands described in this document are available and function only in certain configuration modes on the router.

Information About Managing Configuration Files

Types of Configuration Files

Configuration files contain the Cisco IOS 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 software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

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. As you use the Cisco IOS 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 a Configuration File from a TFTP Server to the Router” section for more information).

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 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 a Configuration File from a TFTP Server to the Router” section for more information.

Configuration File Changes Using the CLI

The Cisco IOS 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 File Transfer Protocol (FTP), remote copy protocol (rcp), or Trivial File Transfer Protocol (TFTP) server. When you configure the software using the CLI, the software executes the commands as you enter them.

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. The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
 - **nvr**am: (NVRAM)

Copy Configuration Files from a Network Server to the Router

You can copy configuration files from a TFTP, rcp, or FTP 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 {ftp | rcp | 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 ftp | rcp | 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:

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

Copy 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.

Copy a Configuration File from the Router to an FTP Server

You can copy a configuration file from the router to an FTP server.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username @routername .domain* . The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** EXEC command if you want to specify a username for that copy operation only.

**Note**

The password must not contain the special characters '@', ':', and '/'. If these special characters are used, the copy fails to parse the IP address of the server.

Copying Files Through a VRF

You can copy files through a VRF interface specified in the **copy** command. Specifying the VRF in the **copy** command is easier and more efficient because you can directly change the source interface without having the need to change the configuration via a change request.

The following example shows how to copy files through a VRF, using the **copy** command:

```

Device# copy scp: slot0: vrf test-vrf
Device# copy scp: slot0: vrf test-vrf
Address or name of remote host [10.1.2.3]?
Source username [ScpUser]?
Source filename [/auto/tftp-server/ScpUser/vrf_test.txt]?
Destination filename [vrf_test.txt]?
Getting the vrf name as test-vrf
Password:
Sending file modes: C0644 10 vrf_test.txt
!
223 bytes copied in 22.740 secs (10 bytes/sec)

```

Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds size of NVRAM, you should be aware of the information in the following sections:

Compressing the Configuration File

The **servicecompress-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 **morenvram: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.

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 **servicecompress-config** global configuration command works only if you have Cisco IOS software Release 10 or later release boot ROMs. Installing new ROMs is a one-time operation and is necessary only if you do not already have Cisco IOS Release 10 in ROM. 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
```

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 USB memory.

See the “Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems ” section for more information.

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.

Loading the Configuration Commands from the Network

You can also store large configurations on FTP or TFTP servers and download them at system startup. To use a network server to store large configurations, see the “Copying a Configuration File from the Router to

a TFTP Server” and “Configure the Router to Download Configuration Files” sections for more information on these commands.

Control of the Parser Cache

The Cisco IOS command-line parser in the Cisco IOS 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 on all platforms using Cisco IOS Release 12.1(5)T and later releases. 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.)

There are several ways to control the parser cache (these are all optional):

- 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
- Disabling the Parser Cache--The Parser Cache feature is enabled by default. To disable the Parser Cache feature, use the `no parser cache` command in global configuration mode. When the parser cache is disabled, the **`noparsercache`** command line is written to the running configuration file. If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **`noparsercache`** command. You will not be able to clear the parser cache after disabling it.
- Reenabling the Parser Cache--To reenab the Parser Cache feature after disabling it, use the **`parsercache`** command in global configuration mode
- 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”).

Configure 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, rcp, or FTP, and must be readable.

How to Manage Configuration File Information

Displaying Configuration File Information

To display information about configuration files, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **show boot**
3. **more *file-url***
4. **show running-config**
5. **show startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	show boot Example: Device# show boot	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.
Step 3	more <i>file-url</i> Example: Device# more 10.1.1.1	Displays the contents of a specified file.
Step 4	show running-config Example: Device# show running-config	Displays the contents of the running configuration file. (Command alias for the more system:running-config command.)

	Command or Action	Purpose
Step 5	show startup-config Example: <pre>Device# show startup-config</pre>	<p>Displays the contents of the startup configuration file. (Command alias for the more nvram:startup-config command.)</p> <p>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.</p>

Modifying the Configuration File at the CLI

The Cisco IOS 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 File Transfer Protocol (FTP), remote copy protocol (rcp), or 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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
 - **end**
 - **^Z**
4. **copy system:running-config nvram:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	<p>Enters global configuration mode. Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.</p>

	Command or Action	Purpose
Step 3	Do one of the following: <ul style="list-style-type: none"> • end • ^Z Example: Device(config)# end	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	copy system:running-config nvram:startup-config Example: Device# 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).

Examples

In the following example, the Device prompt name of the Device is configured. The comment line, indicated by the exclamation mark (!), does not execute any command. The **hostname** command is used to change the Device name from Device 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.

```
Device# configure terminal
Device(config)# !The following command provides the Device host name.
Device(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 Device after rebooting.

Copying a Configuration File from the Router to a TFTP Server

To copy configuration information on a TFTP network server, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **copy system:running-config tftp:** [[[//location]/directory]/filename]
3. **copy nvram:startup-config tftp:** [[[//location]/directory]/filename]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	copy system:running-config tftp: [[[//location]/directory]/filename] Example: <pre>Device# copy system:running-config tftp: //server1/topdir/file10</pre>	Copies the running configuration file to a TFTP server.
Step 3	copy nvram:startup-config tftp: [[[//location]/directory]/filename] Example: <pre>Device# copy nvram:startup-config tftp: //server1/lstdir/file10</pre>	Copies the startup configuration file to a TFTP server.

Examples

The following example copies a configuration file from a Device 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]
```

What to Do Next

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 **fileprompt** global configuration command.

Copying a Configuration File from the Router to the FTP Server

To copy a startup configuration file or a running configuration file from the router to an FTP server, complete the following tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. Do one of the following:
 - **copy system:running-config ftp:** [[[//[*username* [:*password*]@]*location/directory*]/*filename*]
 -
 -
 - **copy nvram:startup-config ftp:** [[[//[*username* [:*password*]@]*location/directory*]/*filename*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip ftp username <i>username</i> Example: Device(config)# ip ftp username user1	(Optional) Specifies the default remote username.
Step 4	ip ftp password <i>password</i> Example: Device(config)# ip ftp password guessme	(Optional) Specifies the default password.
Step 5	end Example: Device(config)# end	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 6	Do one of the following: <ul style="list-style-type: none"> • copy system:running-config ftp: [[[//[<i>username</i> [:<i>password</i>]@]<i>location/directory</i>]/<i>filename</i>] • • 	Copies the running configuration or startup configuration file to an FTP server.

	Command or Action	Purpose
	<ul style="list-style-type: none"> • copy nvram:startup-config ftp: [username[:password]@]location/directory[/filename] <p>Example:</p> <pre>Device# copy system:running-config ftp: //user1:guessme@company.com /dir10/file1</pre>	

Examples

Storing a Running Configuration File on an FTP Server

The following example copies the running configuration file named rtr2-config to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Device# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/Rtr2-config
Write file rtr2-config on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

Storing a Startup Configuration File on an FTP Server

The following example shows how to store a startup configuration file on a server by using FTP to copy the file:

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username netadmin2
Rtr2(config)# ip ftp password mypass
Rtr2(config)# end
Rtr2# copy nvram:startup-config ftp:
Remote host[]? 172.16.101.101
Name of configuration file to write [rtr2-config]?
Write file rtr2-config on host 172.16.101.101?[confirm]
![OK]
```

What to Do Next

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 **fileprompt** global configuration command.

Copying a Configuration File from a TFTP Server to the Router

To copy a configuration file from a TFTP server to the Device, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **copy tftp:** [[[//location]/directory]/filename] **system:running-config**
3. **copy tftp:** [[[//location]/directory]/filename] **nvrn:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	copy tftp: [[[//location]/directory]/filename] system:running-config Example: <pre>Device# copy tftp://server1/dir10/datasource system:running-config</pre>	Copies a configuration file from a TFTP server to the running configuration.
Step 3	copy tftp: [[[//location]/directory]/filename] nvrn:startup-config Example: <pre>Device# copy tftp://server1/dir10/datasource nvrn:startup-config</pre>	Copies a configuration file from a TFTP server to the startup configuration.

Examples

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

```
Device1# 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]
```

What to Do Next

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 **fileprompt** global configuration command.

Copying a Configuration File from an FTP Server to the Router

To copy a configuration file from an F TP server to the running configuration or startup configuration, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. **copy ftp:** [[[//[*username* [:*password*]@]*location*]/*directory*]/*filename*]**system:running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	(Optional) Allows you to enter global configuration mode. This step is required only if you want to override the default remote username or password (see Steps 2 and 3).
Step 3	ip ftp username <i>username</i> Example: <pre>Device(config)# ip ftp username user1</pre>	(Optional) Specifies the default remote username.
Step 4	ip ftp password <i>password</i> Example: <pre>Device(config)# ip ftp password guessme</pre>	(Optional) Specifies the default password.
Step 5	end Example: <pre>Device(config)# end</pre>	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 6	copy ftp: [[[//[<i>username</i> [: <i>password</i>]@] <i>location</i>]/ <i>directory</i>]/ <i>filename</i>] system:running-config Example: Example: <pre>or</pre> Example: <pre>copy ftp: [[[//[<i>username</i> [:<i>password</i>]<i>location</i>/<i>directory</i></pre>	Using FTP, copies the configuration file from a network server to running memory or the startup configuration.

	Command or Action	Purpose
	<pre>] /filename] nvram:startup-config Example: Device# copy ftp://user1:guessme@company.com /dir10/datasource nvram:startup-config</pre>	

Examples

Copy FTP Running-Config

The following example copies a host configuration file named `host1-config` from the `netadmin1` directory on the remote server with an IP address of `172.16.101.101`, and loads and runs those commands on the Device:

```
Device# copy rcp://netadmin1:mypass@172.16.101.101/host1-config system:running-config

Configure using host1-config from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-config:[OK]
Device#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

Copy FTP Startup-Config

The following example specifies a remote username of `netadmin1`. Then it copies the configuration file named `host2-config` from the `netadmin1` directory on the remote server with an IP address of `172.16.101.101` to the startup configuration.

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username
netadmin1
Rtr2(config)# ip ftp password
mypass
Rtr2(config)# end
Rtr2# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-config]? host2-config
Configure using host2-config from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-config:[OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from 172.16.101.101
```

What to Do Next

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 **fileprompt** global configuration command.

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

To compress configuration files, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service compress-config**
4. **end**
5. Do one of the following:
 - Use FTP, rcp, or TFTP to copy the new configuration.
 - **configure terminal**
6. **copy system:running-config nvram:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	service compress-config Example: <pre>Device(config)# service compress-config</pre>	Specifies that the configuration file be compressed.
Step 4	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode.
Step 5	Do one of the following: <ul style="list-style-type: none"> • Use FTP, rcp, or TFTP to copy the new configuration. • configure terminal Example:	Enters the new configuration: <ul style="list-style-type: none"> • If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed:

	Command or Action	Purpose
	Device# <code>configure terminal</code>	“[buffer overflow - <i>file-size</i> / <i>buffer-size</i> bytes].”
Step 6	copy system:running-config nvram:startup-config Example: Device(config)# <code>copy system:running-config nvram:startup-config</code>	When you have finished changing the running-configuration, saves the new configuration.

Examples

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

```
Device# configure terminal

Device(config)# service compress-config

Device(config)# end

Device# 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]
Device# copy system:running-config nvram:startup-config

Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

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

To store the startup configuration in Flash memory, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **copy nvram:startup-config *flash-filesystem:filename***
3. **configure terminal**
4. **boot config flash-filesystem: *filename***
5. **end**
6. Do one of the following:
 - Use FTP, rcp, or 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] .”
 - **configure terminal**
7. **copy system:running-config nvram:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	copy nvram:startup-config flash-filesystem:filename Example: <pre>Device# copy nvram:startup-config slot0:Device-config</pre>	Copies the current startup configuration to the new location to create the configuration file. Note Slot 0 is not applicable for Cisco NCS 4200 Series. Use bootflash or USB.
Step 3	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 4	boot config flash-filesystem: filename Example: <pre>Device(config)# boot config slot0:router-config</pre>	Specifies that the startup configuration file be stored in Flash memory by setting the CONFIG_FILE variable. Note Slot 0 is not applicable for Cisco NCS 4200 Series. Use bootflash or USB.
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode.
Step 6	Do one of the following: <ul style="list-style-type: none"> • Use FTP, rcp, or 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] .” • configure terminal Example: <pre>Device# configure terminal</pre>	Enters the new configuration.
Step 7	copy system:running-config nvram:startup-config Example: <pre>Device(config)# copy system:running-config nvram:startup-config</pre>	When you have finished changing the running-configuration, saves the new configuration.

Examples

The following example stores the configuration file in slot 0:

```
Device# copy nvram:startup-config slot0:router-config

Device# configure terminal

Device(config)# boot config slot0:router-config

Device(config)# end

Device# copy system:running-config nvram:startup-config
```



Note Slot 0 is not applicable for Cisco NCS 4200 Series. Use bootflash or USB.

Loading the Configuration Commands from the Network

To use a network server to store large configurations, complete the tasks in this section:

SUMMARY STEPS

1. enable
2. copy system:running-config {ftp: | rcp: | tftp:}
3. configure terminal
4. boot network {ftp:[[[//[username [:password]@]location]/directory]/filename] | rcp:[[[//[username@]location]/directory]/filename] | tftp:[[[//[location]/directory]/filename]]}
5. service config
6. end
7. copy system:running-config nvram:startup-config

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	copy system:running-config {ftp: rcp: tftp:} Example: <pre>Device# copy system:running-config ftp:</pre>	Saves the running configuration to an FTP, rcp, or TFTP server.
Step 3	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
Step 4	boot network { ftp :[[[//[username [:password]@]location]/directory]/filename] rcp :[[[//[username@]location]/directory]/filename] tftp :[[[//[location]/directory]/filename]} Example: Device(config)# boot network ftp://user1:guessme@company.com/dir10/file1	Specifies that the startup configuration file be loaded from the network server at startup.
Step 5	service config Example: Device(config)# service config	Enables the Device to download configuration files at system startup.
Step 6	end Example: Device(config)# end	Exits global configuration mode.
Step 7	copy system:running-config nvram:startup-config Example: Device# copy system:running-config nvram:startup-config	Saves the configuration.

Managing the Parser Cache

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

Clearing the Parser Cache

To clear the information stored by the Parser Cache feature, complete the task in this section:

SUMMARY STEPS

1. enable
2. clear parser cache

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	clear parser cache Example: Device# clear parser cache	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, complete the task in this section:

SUMMARY STEPS

1. enable
2. configure terminal
3. no parser cache

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	no parser cache Example: Device(config)# no parser cache	Disables the Parser Cache feature: <ul style="list-style-type: none"> • When the parser cache is disabled, the noparsercache command line is written to the running configuration file. • If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the noparsercache 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, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parser cache**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	parser cache Example: <pre>Device(config)# parser cache</pre>	Enables the Parser Cache feature.

What to Do Next

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.

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 of the commands in Step 2:

SUMMARY STEPS

1. **enable**
2. Do one of the following:
 - **copy filesystem : [partition-number:][filename] nvram:startup-config**
 - **copy filesystem : [partition-number:][filename] system:running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Do one of the following: <ul style="list-style-type: none"> • copy filesystem : [partition-number:][filename] nvram:startup-config • copy filesystem : [partition-number:][filename] system:running-config Example: <pre>Device# copy slot0:4:ios-upgrade-1 nvram:startup-config</pre>	Loads a configuration file directly into NVRAM. or Copies a configuration file to your running configuration. Note Slot 0 is not applicable for Cisco NCS 4200 Series. Use bootflash or USB.

Examples

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

```
Device# copy slot0:4:ios-upgrade-1 nvram:startup-config

Copy '
ios-upgrade-1
' from flash device
  as 'startup-config' ? [yes/no] yes

[OK]
```



Note Slot 0 is not applicable for Cisco NCS 4200 Series. Use bootflash or USB.

Copying a Configuration File from an FTP Server to Flash Memory Devices

To copy a configuration file from an FTP server to a Flash memory device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**

6. copy ftp: [[[/[username:password@]location]/directory]/filename]
flash-filesystem:[partition-number:][filename]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 3	ip ftp username username Example: Device(config)# ip ftp username user1	(Optional) Specifies the remote username.
Step 4	ip ftp password password Example: Device(config)# ip ftp password guessme	(Optional) Specifies the remote password.
Step 5	end Example: Device(config)# end	(Optional) Exits configuration mode. This step is required only if you override the default remote username (see Steps 3 and 4).
Step 6	copy ftp: [[[/[username:password@]location]/directory]/filename] flash-filesystem:[partition-number:][filename] Example: Device> copy ftp:router-config slot0:new-config	Copies the configuration file from a network server to the Flash memory device using FTP.

What to Do Next

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 **fileprompt** global configuration command.

Copying a Configuration File from an rcp Server to Flash Memory Devices

To copy a configuration file from an rcp server to a Flash memory device, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username *username***
4. **end**
5. **copy rcp: [[[//[*username*@]*location*]/*directory*]/*filename*] flash-filesystem:[*partition-number*:]*filename*]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 3	ip rcmd remote-username <i>username</i> Example: <pre>Device(config)# ip rcmd remote-username user1</pre>	(Optional) Specifies the remote username.
Step 4	end Example: <pre>Device(config)# end</pre>	(Optional) Exits configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 5	copy rcp: [[[//[<i>username</i>@]<i>location</i>]/<i>directory</i>]/<i>filename</i>] flash-filesystem:[<i>partition-number</i>:]<i>filename</i>] Example: <pre>Device# copy rcp:router-config slot0:new-config</pre>	Copies the configuration file from a network server to the Flash memory device using rcp. Reply to any router prompts for additional information or confirmation. The prompting will depend on how much information you provide in the copy command and the current setting of the fileprompt command.

Copying a Configuration File from a TFTP Server to Flash Memory Devices

To copy a configuration file from a TFTP server to a Flash memory device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **copy tftp: [[[//[*location*]/*directory*]/*filename*] flash-filesystem:[*partition-number*:]*filename*]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	copy tftp: [[//location]/directory]/filename] flash-filesystem:[partition-number:]/filename] Example: <pre>Device# copy tftp:router-config slot0:new-config</pre>	Copies the file from a TFTP server to the Flash memory device. Reply to any Device prompts for additional information or confirmation. The prompting will depend on how much information you provide in the copy command and the current setting of the fileprompt command.

Examples

The following example shows the copying of the configuration file named Device-config from a TFTP server to the Flash memory card inserted in slot 0 of the Network Processing Engine (NPE) or Route Switch Processor (RSP) card of a Cisco 7500 series Device. The copied file is renamed new-config.

```
Device# copy tftp:router-config slot0:new-config
```

Reexecuting the Configuration Commands in the Startup Configuration File

To reexecute the commands located in the startup configuration file, complete the task in this section:

SUMMARY STEPS

1. enable
2. configure memory

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure memory Example: <pre>Device# configure memory</pre>	Reexecutes the configuration commands located in the startup configuration file .

Clearing the Startup Configuration

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. To clear the contents of your startup configuration, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **erase nvram**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	erase nvram Example: <pre>Device# erase nvram</pre>	Clears the contents of your startup configuration. Note 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 erasestartup-configEXEC command, the Device erases or deletes the configuration pointed to by CONFIG_FILE environment variable. If this variable points to NVRAM, the Device erases NVRAM. If the CONFIG_FILE environment variable specifies a Flash memory device and configuration filename, the Device deletes the configuration file. That is, the Device 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, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **delete flash-filesystem : filename**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	delete <i>flash-filesystem</i> : <i>filename</i> Example: <pre>Device# delete slot0:myconfig</pre>	Deletes a specified configuration file on a specified Flash device. Note On Class A and B Flash file systems, when you delete a specific file in Flash memory, the system marks the file as deleted, allowing you to later recover a deleted file using the undelete EXEC command. Erased files cannot be recovered. To permanently erase the configuration file, use the squeeze EXEC command. On Class C Flash file systems, you cannot recover a file that has been deleted. If you attempt to erase or delete the configuration file specified by the CONFIG_FILE environment variable, the system prompts you to confirm the deletion.

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems

On Class A Flash file systems, you can configure the Cisco IOS 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, complete the tasks in this section:

SUMMARY STEPS

1. enable
2. copy *[flash-url|ftp-url|rcp-url|tftp-url|system:running-config|nvram:startup-config] dest-flash-url*
3. configure terminal
4. boot config *dest-flash-url*
5. end
6. copy system:running-config nvram:startup-config
7. show boot

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
	<code>enable</code>	
Step 2	copy <i>[flash-url ftp-url rcp-url tftp-url system:running-config nvram:startup-config]</i> <i>dest-flash-url</i> Example: <pre>Device# copy system:running-config nvram:startup-config</pre>	Copies the configuration file to the Flash file system from which the router will load the file upon restart.
Step 3	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 4	boot config <i>dest-flash-url</i> Example: <pre>Device(config)# boot config 172.16.1.1</pre>	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode.
Step 6	copy system:running-config nvram:startup-config Example: <pre>Device# copy system:running-config nvram:startup-config</pre>	Saves the configuration performed in Step 3 to the startup configuration.
Step 7	show boot Example: <pre>Device# show boot</pre>	(Optional) Allows you to verify the contents of the CONFIG_FILE environment variable.

Examples

The following example copies the running configuration file to the first PCMCIA slot of the RSP card in a Cisco 7500 series router. This configuration is then used as the startup configuration when the system is restarted.

```
Device# copy system:running-config slot0:config2
Device# configure terminal
Device(config)# boot config slot0:config2
Device(config)# end
Device# copy system:running-config nvram:startup-config
[ok]
```

```

Device# show bootvar
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:config2
Configuration register is 0x010F

```

What to Do Next

After you specify a location for the startup configuration file, the **nvrnram:startup-config** command is aliased to the new location of the startup configuration file. The **morenvram:startup-config** EXEC command will display the startup configuration, regardless of its location. The **erasenvram: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 **copysystem:running-confignvram: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 **copysystem:running-confignvram: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.

Configuring the Router to Download the Network Configuration File

To configure the Cisco IOS software to download a network configuration file from a server at startup, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot network** {ftp:[[/[/[username [:password]@]location]/directory]/filename] | rcp:[[/[/[username@]location]/directory]/filename] | tftp:[[/[/[location]/directory]/filename]}
4. **service config**
5. **end**
6. Device# **copy system:running-config nvram:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	boot network {ftp:[[/[/username[:password]@]location]/directory]/filename] rcp:[[/[/username@]location]/directory]/filename] tftp:[[/[/location]/directory]/filename]} Example: Device(config)# boot network tftp:hostfile1	Specifies the network configuration file to download at startup, and the protocol to be used (TFTP, rcp, or FTP). <ul style="list-style-type: none"> • If you do not specify a network configuration filename, the Cisco IOS software uses the default filename network-config. If you omit the address, the Device 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.
Step 4	service config Example: Device(config)# service config	Enables the system to automatically load the network file upon restart.
Step 5	end Example: Device(config)# end	Exits global configuration mode.
Step 6	Device# copy system:running-config nvram:startup-config Example: Device# copy system:running-config nvram:startup-config	Saves the running configuration to the startup configuration file.

Configuring the Router to Download the Host Configuration File

To configure the Cisco IOS software to download a host configuration file from a server at startup, complete the tasks in this section:

SUMMARY STEPS

1. enable
2. configure terminal

3. **boot host** {**ftp:**[[[//[username [:password]@]location]/directory]/filename] | **rcp:**[[[//[username@]location]/directory]/filename] | **tftp:**[[[//[location]/directory]/filename] }
4. **service config**
5. **end**
6. **copy system:running-config nvram:startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	boot host { ftp: [[[//[username [:password]@]location]/directory]/filename] rcp: [[[//[username@]location]/directory]/filename] tftp: [[[//[location]/directory]/filename] } Example: <pre>Device(config)# boot host tftp:hostfile1</pre>	Specifies the host configuration file to download at startup, and the protocol to be used (FTP, rcp, or TFTP): <ul style="list-style-type: none"> • If you do not specify a host configuration filename, the Device uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending “-config.” If no host name information is available, the software uses the default host configuration filename Device-config. If you omit the address, the Device uses the broadcast address. • You can specify more than one host configuration file. The Cisco IOS 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.
Step 4	service config Example: <pre>Device(config)# service config</pre>	Enables the system to automatically load the host file upon restart.
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode.
Step 6	copy system:running-config nvram:startup-config Example:	Saves the running configuration to the startup configuration file.

	Command or Action	Purpose
	Device# copy system:running-config nvram:startup-config	

Examples

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

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS configuration commands	Cisco IOS Configuration Fundamentals Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 2

Exclusive Configuration Change Access and Access Session Locking

Exclusive Configuration Change Access (also called the “Configuration Lock” feature) allows you to have exclusive change access to the Cisco IOS XE running configuration, preventing multiple users from making concurrent configuration changes.

The Access Session Locking addition to this feature extends the Exclusive Configuration Change Access feature such that **show** and **debug** commands entered by the user holding the configuration lock always have execution priority; **show** and **debug** commands entered by other users are only allowed to run after the processes initiated by the configuration lock owner have finished.

The Exclusive Configuration Change Access feature (“exposed lock”) is complementary with the locking mechanism in the Configuration Replace and Configuration Rollback feature (“rollback lock”).

- [Finding Feature Information, on page 35](#)
- [Information About Locking the Configuration, on page 36](#)
- [How to Configure Configuration Exclusive Configuration Change-Access and Access Session Locking, on page 37](#)
- [Configuration Examples for Locking the Configuration, on page 40](#)
- [Additional References, on page 41](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Information About Locking the Configuration

Exclusive Configuration Change Access and Access Session Locking

Devices running Cisco IOS software maintain a running configuration that determines the configuration state of the device. Changes to the running configuration alter the behavior of the device. Because Cisco IOS software allows multiple users to change the running configuration via the device CLI (including the device console and telnet Secure Shell (SSH)), in some operating environments it would be beneficial to prevent multiple users from making concurrent changes to the Cisco IOS running configuration. Temporarily limiting access to the Cisco IOS running configuration prevents inadvertent conflicts or cases where two users attempt to configure the same portion of the running configuration.

The Exclusive Configuration Change Access feature (also called the “Configuration Lock” feature) allows you to have exclusive change access to the Cisco IOS running configuration, preventing multiple users from making concurrent configuration changes.

This feature provides exclusive change access to the Cisco IOS running configuration from the time you enter global configuration mode by using the **configure terminal** command. This gives the effect of a “configuration lock,” preventing other users from changing the Cisco IOS running configuration. The configuration lock is automatically released when the user exits Cisco IOS configuration mode.

The Exclusive Configuration Change Access feature is enabled using the **configuration mode exclusive** command in global configuration mode. Exclusive configuration change access can be set to **auto**, so that the Cisco IOS configuration mode is locked whenever anyone uses the **configure terminal** command, or it can be set to **manual**, so that the Cisco IOS configuration mode is locked only when the **configure terminal lock** command is issued.

The Exclusive Configuration Change Access feature is complementary with the locking mechanism for the Configuration Replace and Configuration Rollback feature introduced in Cisco IOS Release 12.2(25)S and 12.3(7)T.

Access Session Locking

The Access Session Locking feature extends the Exclusive Configuration Change Access feature such that **show** and **debug** commands entered by the user holding the configuration lock always have execution priority. This feature prevents concurrent configuration access and also provides an option to prevent simultaneous processes, such as a **show** command entered by another user, from executing while other configuration commands are being executed. When this feature is enabled, the commands entered by the user with the configuration lock (such as configuration commands) always have priority over commands entered by other users.

How to Configure Configuration Exclusive Configuration Change-Access and Access Session Locking

Enabling Exclusive Configuration Change Access and Access Session Locking



Note Effective with Cisco IOS Release 12.2(33)SRE, the Exclusive Configuration Change Access and Access Session Locking feature is not available in Cisco IOS software. Use the Parser Concurrency and Locking Improvements feature instead of this feature. See the “Enabling Parser Concurrency and Locking Improvements” section for more information.

Perform this task to enable the Exclusive Configuration Change Access and Access Session Locking feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configuration mode exclusive**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	configuration mode exclusive Example: Router(config)# configuration mode exclusive	Enables exclusive configuration change access (configuration lock feature). <ul style="list-style-type: none">• When the command is enabled, configuration sessions are performed in single-user (exclusive) mode.• Execution of this command locks the configuration sessions automatically.
Step 4	end Example:	Ends your configuration session and returns the CLI to privileged EXEC mode.

	Command or Action	Purpose
	Router(config)# end	

Obtaining Exclusive Configuration Change Access

Perform this task to obtain exclusive configuration change access manually for the duration of your configuration session. Use the **lock** keyword with the **configure terminal** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure terminal lock**
4. Configure the system by entering your changes to the running configuration.
5. Do one of the following:
 - **end**
 - **or**
 - **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	configure terminal lock Example: Router(config)# configure terminal lock	(Optional) Locks the Cisco IOS software in exclusive (single-user) mode. <ul style="list-style-type: none"> • This command can be used only if you have previously enabled configuration locking by using the configuration mode exclusive command. <p>Note This command is not required for performing a manual lock.</p> <ul style="list-style-type: none"> • This command is available in Cisco IOS Release 12.3(14)T or later releases.
Step 4	Configure the system by entering your changes to the running configuration.	--

	Command or Action	Purpose
Step 5	<p>Do one of the following:</p> <ul style="list-style-type: none"> • end • or • exit <p>Example:</p> <pre>Router(config)# end</pre> <p>Example:</p> <pre>Router(config)# exit</pre>	<p>Ends your configuration session, automatically releases the session lock obtained in Step 1, and exits to privileged EXEC mode.</p> <p>Note Either the end command, the exit command, or the Ctrl-Z key combination releases the configuration lock. Use of the end command is recommended.</p>

Monitoring and Troubleshooting Configuration Locking

Perform either or both steps in this task to monitor or troubleshoot the Exclusive Configuration Change Access and Access Session Locking feature.

SUMMARY STEPS

1. **show configuration lock**
2. **debug configuration lock**

DETAILED STEPS

Step 1 **show configuration lock**

Use this command to display the status and details of any current configuration locks, including the owner, user, terminal, lock state, and lock class.

If you cannot enter global configuration mode, you can use this command to determine if the configuration session is locked by another user, and who that user is.

Example:

```
Router# show configuration lock
Config Session Lock
-----
Owner PID : 543
TTY number : 2
TTY username : unknown
User debug info : CLI Session Lock
Lock Active time (in Sec) : 63
Router(config)#
```

Step 2 **debug configuration lock**

Use this command to enable debugging of Cisco IOS configuration locks (exposed class locks or rollback class locks):

Example:

```

Router# debug configuration lock

Session1 from console
=====
Router# configure terminal lock
Configuration mode locked exclusively. The lock will be cleared once you exit out of configuration
mode using end/exit
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Parser : LOCK REQUEST in EXCLUSIVE mode
Parser: <configure terminal lock> - Config. Lock requested by process <3> client <PARSER Client>
Parser: <configure terminal lock> - Config. Lock acquired successfully !
Router(config)#

```

Configuration Examples for Locking the Configuration

Configuring an Exclusive Lock in Auto Mode Example

The following example shows how to enable the exclusive lock in auto mode for single-user auto configuration mode using the **configurationmodeexclusive** command. Once the Cisco IOS configuration file is locked exclusively, you can verify this configuration by using the **showconfigurationlock** command.

```

Router# configure terminal
Router(config)# configuration mode exclusive
Router(config)# exit
Router# configure terminal
! Locks configuration mode exclusively.
Router# show configuration lock
Parser Configure Lock
Owner PID      : 10
User           : User1
TTY            : 3
Type           : EXCLUSIVE
State          : LOCKED
Class          : Exposed
Count          : 0
Pending Requests : 0
User debug info : 0

```

Configuring an Exclusive Lock in Manual Mode Example

The following example shows how to enable the exclusive locking feature in manual mode by using the **configure terminal lock** command. Use the **configure terminal lock** command to lock manually. In this mode, the **configure terminal** command will not automatically lock the parser configuration mode.

```

Router# configure terminal
Router# configure terminal lock
Enter configuration commands, one per line. End with CNTL/Z.
*Mar 25 17:02:45.928: Configuration mode locked exclusively. The lock will be cleared once
you exit out of configuration mode using end/exit

```

Additional References

The following sections provide references related to locking the configuration.

Related Documents

Related Topic	Document Title
Commands for managing configuration files	<i>Cisco IOS Configuration Management Command Reference</i>
Information about managing configuration files	<i>Managing Configuration Files</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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



CHAPTER 3

Configuration Partitioning

The Configuration Partitioning feature provides modularization (“partitioning”) of the running configuration state to provide granular access to the running configuration in Cisco IOS software.

This feature is enabled by default in Cisco IOS software images that include this feature.

The configuration state of a device is retrieved dynamically whenever a user issues the **showrunning-config** command. When the Configuration Partitioning feature is enabled, the system groups the configuration state of the device into parts (called “partitions”) so that only the configuration state the user wishes to review is retrieved when generating a displayed list of commands in the running configuration. This feature improves performance for high-end systems with complex configurations because only a part of the running configuration state is processed when generating the running configuration command list, as opposed to the existing method of processing the entire system configuration state.

Default configuration partitions are provided by the introduction of this feature; other Cisco IOS software features may define their own command partitions in later releases.

- [Finding Feature Information, on page 43](#)
- [Information About Configuration Partitioning, on page 44](#)
- [How to Use the Configuration Partitioning Feature, on page 45](#)
- [Configuration Examples for Configuration Partitioning, on page 48](#)
- [Additional References, on page 57](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Information About Configuration Partitioning

System Running Configurations

Managing the configuration of any Cisco IOS software-based device involves managing the startup configuration (startup-config), which is a file stored in nonvolatile memory, and the running configuration (running-config), which is the set of all configuration options currently in effect on the system. Typically, the startup configuration file is loaded when the system boots, and changes to the system's running configuration, applied using the command-line interface (CLI), are saved by copying the running configuration to a configuration file (either locally or on the network), which can then be used to configure the device at startup, or used to configure other devices.

Retrieving the Running Configuration for Display or Copy Operations

In the Cisco IOS software configuration model, the configuration state is maintained in a distributed manner, with each component storing its own configuration state. To retrieve global configuration information, the software must poll every component to collect the distributed information. This configuration state retrieval operation is performed by a process known as nonvolatile generation (NVGEN), and it is invoked by commands such as **showrunning-config**, which is used to display the current configuration state, and **copysystem:running-configuration**, which is used to save the running configuration by copying it to a file. When invoked, the NVGEN process queries each system component, each interface instance, and all other configured component objects in a standard sequence. A running configuration file is constructed as NVGEN traverses the system performing these queries, and it is this "virtual file" that is displayed or copied.

Benefits of Partitioning the Running Configuration

The Configuration Partitioning feature is the latest in a series of Configuration Generation Performance Enhancement Features for Cisco IOS software. (See the "Related Documents" section for related features.) This feature improves the system's response time by providing a method for querying only the system component you wish to review when issuing the **showrunning-config** command.

When the Configuration Partitioning feature is enabled, the system groups the configuration state of the device into parts (called "partitions") for the purpose of generating the virtual running configuration file (the list of configuration commands). A new command, **showrunning-configpartition**, allows you to display only the part of the running configuration that you want to examine, rather than having to display the entire running configuration at once, or displaying only lines that match a certain string.

The key benefit of this feature is that it increases system performance by allowing the system to run the NVGEN process for only the collection of system components (such as specific interfaces) that you need to display. This is in contrast to other existing extensions to the **showrunning-config** command, which only filter the generated list after all system components have been processed.

The selective processing of the system's configuration state for the purpose of generating a partial running configuration is called "configuration partitioning."

More granular access to configuration information offers important performance benefits for high-end routing platforms with very large configuration files, while also enhancing configuration management by allowing advanced configuration features to be implemented at a more granular level. Advanced configuration options

include Cisco IOS software support for provisioning of customer services, Config Rollback, Config Locking, and configuration access control.

How to Use the Configuration Partitioning Feature

Displaying Configuration Partitions

The main method of taking advantage of this feature is by using the **showrunning-configpartition***part* command in priveledge exec mode, which is a specialized extension to the **showrunning-config** command.



Note The **partition***part* command extension is not available for the **more:systemrunning-config** command.

Because this feature offers improved performance for existing commands, this feature is enabled by default in Cisco IOS software images that support this feature. To quickly determine if this feature is supported and running on your system, issue the **showrunning-configpartition?** command in privileged EXEC mode.

SUMMARY STEPS

1. **show running-config partition ?**
2. **show running-config partition** *part*

DETAILED STEPS

Step 1 **show running-config partition ?**

Issuing this command will show you the list of running configuration parts available for display on your system.

If the Configuration Partitioning feature is supported on your system and is enabled, you will see the string “ config partition is TRUE ” as the first line of help output.

If you receive an error message when entering the command syntax shown here, this feature is not supported on your system. See the command documentation for the **showrunning-config** command for existing extensions of that command in other releases that allow you to show only part of the running configuration.

Note The list of available configuration parts may vary by software image and is dependent on what features are currently configured.

Example:

```
Router# show running-config partition ?
config partition is TRUE
access-list      All access-list configurations
boot             All boot configurations
class-map        All class-map configurations
common           All remaining unregistered configurations
global-cdp       All global cdp configurations
interface        All Interface specific Configurations
ip-as-path       All IP as-path configurations
ip-community     All IP community list configurations
ip-domain-list   All ip domain list configurations
```

ip-prefix-list	All ip prefix-list configurations
ip-static-routes	All IP static configurations
line	All line mode configurations
policy-map	All policy-map configurations
route-map	All route-map configurations
router	All routing configurations
snmp	All SNMP configurations
tacacs	All TACACS configurations

Choose the part of the running configuration you want to display, and use the associated keyword as the *part* argument in Step 2.

Step 2 **show running-config partition** *part*

As an example, to have the system perform the NVGEN process on only the components associated with the access-list parts of the running configuration state, and display only the access-list related configurations, you would enter the **showrunning-configpartitionaccess-list** command:

Example:

```
Router# show running-config partition access-list
Building configuration...
Current configuration : 127 bytes
!
Configuration of Partition access-list
!
!
!
access-list 90 permit 0.0.0.0 1.2.3.5
access-list 100 permit 10 any any
!
end
```

Note This command also allows you to run the NVGEN process and display the resulting output for specific interfaces. This is a key capability of this feature, as it was designed for systems with numerous active interfaces.

In the following example, the main configuration partition is the interface configuration, and the specific part of the configuration to be generated is the configuration for Fast Ethernet interface 0/0.

Example:

```
Router# show running-config partition interface fastethernet0/0
Building configuration...
Current configuration : 213 bytes
!
Configuration of Partition interface FastEthernet0/0
!
!
!
interface FastEthernet0/0
 ip address 10.4.2.39 255.255.255.0
 no ip route-cache cef
 no ip route-cache
 duplex half
 ipv6 enable
 no cdp enable
!
!
end
```

Disabling the Configuration Partitioning Feature

Because this feature offers improved performance for existing commands, this feature is enabled by default for Cisco IOS software images that support this feature. However, you may want to disable this feature if you determine that it is not needed, as this feature does use a small amount of system resources (memory and CPU utilization). To disable configuration partitioning, perform the following task, which assumes you are starting in user EXEC mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no parser config partition**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	no parser config partition Example: Router(config)# no parser config partition Example: Disabling config partitioning Example: Router(config)#	Disables the configuration partitioning feature.

What to do next

What to Do Next

To reenable the feature after it has been disabled, use the **parserconfigpartition** command in global configuration mode.



Note As this feature is enabled by default, only the **no** form will appear in the running configuration file, or will be written to the startup configuration file when you issue the **copyrunning-configstartup-config** command.

Configuration Examples for Configuration Partitioning

This section provides examples of displaying configuration partitions with the **show running-config partition** command:

Displaying Configuration Partitions Example

In this example, the **showrunning-configpartition** command is used with related commands in a series of steps an administrator might take to check the status of a specific interface and the current configuration of some of the system's other components. Comparable filtered output from the standard **showrunning-config** command (for example, **showrunning-configincludeaccess-list**) is included for demonstration purposes.



Note The *part* argument can consist of multiple partition name keywords, as in **showrunning-configpartroutrereigrp1**.

```
gt3-7200-3# show running-config partition ?
access-list      All access-list configurations
boot             All boot configurations
class-map        All class-map configurations
global-cdp       All global cdp configurations
interface        All Interface specific Configurations
ip-as-path       All IP as-path configurations
ip-community     All IP community list configurations
ip-domain-list   All ip domain list configurations
ip-static-routes All IP static configurations
line             All line mode configurations
policy-map       All policy-map configurations
route-map        All route-map configurations
router           All routing configurations
service          All service configurations
snmp             All SNMP configurations
gt3-7200-3# show running-config partition access-list

Building configuration...
Current configuration : 87 bytes
!
!
!
!
access-list 90 permit 0.0.0.0 1.2.3.5
access-list 100 permit 10 any any
!
end
gt3-7200-3# show running-config | include access-list

access-list 90 permit 0.0.0.0 1.2.3.5
```

```

access-list 100 permit 10 any any
gt3-7200-3#
gt3-7200-3# show running-config partition boot

Building configuration...
Current configuration : 51 bytes
!
boot network tftp:/service_config.txt
!
!
!
end
gt3-7200-3# show running-config partition class-map

Building configuration...
Current configuration : 78 bytes
!
!
!
class-map match-all abc
  match any
class-map match-all xyz
!
!
!
end
gt3-7200-3# show running-config | begin class-map

class-map match-all abc
  match any
class-map match-all xyz
!
!
gt3-7200-3# show running-config partition global-cdp

Building configuration...
Current configuration : 43 bytes
!
!
!
cdp timer 20
cdp holdtime 100
!
end
gt3-7200-3# show running-config | include

global-cdp

cdp timer 20
cdp holdtime 100
gt3-7200-3#
gt3-7200-3# show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	NVRAM	administratively down	down
Ethernet2/0	10.4.2.32	YES	NVRAM	up	up
Ethernet2/1	unassigned	YES	NVRAM	administratively down	down
Ethernet2/2	unassigned	YES	NVRAM	administratively down	down
Ethernet2/3	unassigned	YES	NVRAM	administratively down	down
Serial3/0	unassigned	YES	NVRAM	administratively down	down
Serial3/1	unassigned	YES	NVRAM	administratively down	down
Serial3/2	unassigned	YES	NVRAM	administratively down	down
Serial3/3	unassigned	YES	NVRAM	administratively down	down
Loopback0	unassigned	YES	NVRAM	administratively down	down
Loopback234	unassigned	YES	NVRAM	administratively down	down

```

gt3-7200-3# show running-config partition interface fastethernet0/0
Building configuration...
Current configuration : 98 bytes
!
!
!
interface FastEthernet0/0
  no ip address
  no ip route-cache
  shutdown
  duplex half
!
!
end
gt3-7200-3# show running-config partition interface ethernet2/0

Building configuration...
Current configuration : 122 bytes
!
!
!
interface Ethernet2/0
  ip address 10.4.2.32 255.255.255.0
  no ip proxy-arp
  no ip route-cache
  duplex half
!
!
end
gt3-7200-3# show running-config partition interface ethernet2/1
Building configuration...
Current configuration : 94 bytes
!
!
!
interface Ethernet2/1
  no ip address
  no ip route-cache
  shutdown
  duplex half
!
!
end
gt3-7200-3# show running-config partition interface ethernet2/2

Building configuration...
Current configuration : 94 bytes
!
!
!
interface Ethernet2/2
  no ip address
  no ip route-cache
  shutdown
  duplex half
!
!
end
gt3-7200-3# show running-config partition interface ethernet2/3
Building configuration...
Current configuration : 94 bytes
!
!
!

```



```
interface Ethernet2/3
  no ip address
  no ip route-cache
  shutdown
  duplex half
!
!
end
gt3-7200-3# show running-config partition interface serial3/0
Building configuration...
Current configuration : 103 bytes
!
!
!
interface Serial3/0
  no ip address
  no ip route-cache
  shutdown
  serial restart-delay 0
!
!
end
gt3-7200-3# show running-config partition interface serial3/1
Building configuration...
Current configuration : 103 bytes
!
!
!
interface Serial3/1
  no ip address
  no ip route-cache
  shutdown
  serial restart-delay 0
!
!
end
gt3-7200-3# show running-config partition interface serial3/2
Building configuration...
Current configuration : 103 bytes
!
!
!
interface Serial3/2
  no ip address
  no ip route-cache
  shutdown
  serial restart-delay 0
!
!
end
gt3-7200-3# show running-config partition interface serial3/3
Building configuration...
Current configuration : 103 bytes
!
!
!
interface Serial3/3
  no ip address
  no ip route-cache
  shutdown
  serial restart-delay 0
!
!
end
```

```

gt3-7200-3# show running-config partition interface loopback0
Building configuration...
Current configuration : 79 bytes
!
!
!
interface Loopback0
  no ip address
  no ip route-cache
  shutdown
!
!
end
gt3-7200-3# show running-config partition interface loopback1
                                     ^
% Invalid input detected at '^' marker.
gt3-7200-3# show running-config partition interface loopback234
Building configuration...
Current configuration : 81 bytes
!
!
!
interface Loopback234
  no ip address
  no ip route-cache
  shutdown
!
!
end
gt3-7200-3# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
gt3-7200-3(config)# interface ethernet 2/0.1
gt3-7200-3(config-subif)# exit
gt3-7200-3(config)# exit
gt3-7200-3#
00:13:05: %SYS-5-CONFIG_I: Configured from console by console
gt3-7200-3# show running-config partition interface ethernet2/0.1
Building configuration...
Current configuration : 58 bytes
!
!
!
interface Ethernet2/0.1
  no ip route-cache
!
!
end
gt3-7200-3# show run partition ip?
ip-as-path ip-community ip-domain-list ip-static-routes
gt3-7200-3#sh run part ip-as
gt3-7200-3#sh run part ip-as-path

Building configuration...
Current configuration : 125 bytes
!
!
!
ip as-path access-list 2 permit $ABC
ip as-path access-list 2 permit $xyz*
ip as-path access-list 2 permit qwe*
!
end
gt3-7200-3# show running-config partition ip-community

```

```
Building configuration...
Current configuration : 92 bytes
!
!
!
ip community-list standard asd permit
ip community-list expanded qwe deny uio*
!
end
gt3-7200-3# show running-config | include ip community
ip community-list standard asd permit
ip community-list expanded qwe deny uio*
gt3-7200-3#
gt3-7200-3# show running-config partition ip-domain-list

Building configuration...
Current configuration : 70 bytes
!
ip domain-list iop
ip domain-list tyu
ip domain-list jkl
!
!
!
end
gt3-7200-3# show running-config partition
ip-static-routes

Building configuration...
Current configuration : 98 bytes
!
!
!
ip route 0.0.0.0 0.0.0.0 Ethernet2/0
ip route 171.69.1.129 255.255.255.255 10.4.29.1
!
end
gt3-7200-3# show running-config partition line
Building configuration...
Current configuration : 489 bytes
!
!
!
!
line con 0
  exec-timeout 0 0
  transport output lat pad v120 mop telnet rlogin udptn nasi
  stopbits 1
line aux 0
  transport output lat pad v120 mop telnet rlogin udptn nasi
  stopbits 1
line vty 0
  password lab
  login
  transport input lat pad v120 mop telnet rlogin udptn nasi
  transport output lat pad v120 mop telnet rlogin udptn nasi
line vty 1 4
  login
  transport input lat pad v120 mop telnet rlogin udptn nasi
  transport output lat pad v120 mop telnet rlogin udptn nasi
!
end
gt3-7200-3# show running-config partition policy-map
```

```

Building configuration...
Current configuration : 162 bytes
!
!
!
policy-map qwer
  description policy-map qwer.
  class xyz
    shape peak 8000 32 32
policy-map p1
policy-map sdf
  class abc
    set precedence 4
!
!
!
end
gt3-7200-3# show running-config partition route-map
Building configuration...
Current configuration : 65 bytes
!
!
!
route-map iop permit 10
!
route-map rty permit 10
!
!
end
gt3-7200-3#sh run part router bgp 1
Building configuration...
Current configuration : 111 bytes
!
!
!
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  distance bgp 2 2 2
  no auto-summary
!
!
end
gt3-7200-3#sh run part router egp ?
<0-65535> Remote autonomous system number
gt3-7200-3#sh run part router egp 1
Building configuration...
Current configuration : 46 bytes
!
!
!
router egp 1
  timers egp 20 20
!
!
end
gt3-7200-3# show running-config partition router ?
  bgp      Border Gateway Protocol (BGP)
  egp      Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      ISO IS-IS
  iso-igrp  IGRP for OSI networks
  mobile    Mobile routes
  odr       On Demand stub Routes

```

```

    ospf      Open Shortest Path First (OSPF)
    rip       Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router eigrp ?
<1-65535> Autonomous system number
gt3-7200-3# show running-config partition router eigrp 1
Building configuration...
Current configuration : 13 bytes
!
!
!
!
end
gt3-7200-3#
gt3-7200-3# sh run part router eigrp 2

Building configuration...
Current configuration : 57 bytes
!
!
!
router eigrp 2
  variance 10
  auto-summary
!
!
end
gt3-7200-3# show running-config partition router ?
  bgp      Border Gateway Protocol (BGP)
  egp      Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      ISO IS-IS
  iso-igrp  IGRP for OSI networks
  mobile    Mobile routes
  odr       On Demand stub Routes
  ospf      Open Shortest Path First (OSPF)
  rip       Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router isis ?
  WORD     ISO routing area tag
  |        Output modifiers
  <cr>
gt3-7200-3# show running-config partition router isis qwe
Building configuration...
Current configuration : 86 bytes
!
!
!
router isis qwe
  set-attached-bit route-map qwer
  use external-metrics
!
!
end
gt3-7200-3# show running-config partition router isis ?
  WORD     ISO routing area tag
  |        Output modifiers
  <cr>
gt3-7200-3# show running-config partition router iso
gt3-7200-3# show running-config partition router iso-igrp ?
  WORD     ISO routing area tag
  |        Output modifiers
  <cr>
gt3-7200-3# show running-config partition router iso-igrp

Building configuration...

```

```

Current configuration : 31 bytes
!
!
!
router iso-igrp
!
!
end
gt3-7200-3# show running-config | begin iso
router iso-igrp
!
router isis qwe
  set-attached-bit route-map qwer
  use external-metrics
!
router egp 1
  timers egp 20 20
!
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  distance bgp 2 2 2
  no auto-summary
!

gt3-7200-3# show running-config partition router ?
  bgp      Border Gateway Protocol (BGP)
  egp      Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      ISO IS-IS
  iso-igrp  IGRP for OSI networks
  mobile    Mobile routes
  odr       On Demand stub Routes
  ospf      Open Shortest Path First (OSPF)
  rip       Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router mobile ?
  | Output modifiers
  <cr>
gt3-7200-3# show running-config partition router mobile

Building configuration...
Current configuration : 42 bytes
!
!
!
router mobile
  distance 20
!
!
end
gt3-7200-3# sh run | include router

router mobile
router odr
router eigrp 2
router ospf 4
router iso-igrp
router isis qwe
router egp 1
router bgp 1
gt3-7200-3# show running-config partition router ?
  bgp      Border Gateway Protocol (BGP)
  egp      Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)

```

```

isis      ISO IS-IS
iso-igrp   IGRP for OSI networks
mobile    Mobile routes
odr        On Demand stub Routes
ospf       Open Shortest Path First (OSPF)
rip        Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router ospf ?
<1-65535>  Process ID
gt3-7200-3# show running-config partition router ospf 4
Building configuration...
Current configuration : 64 bytes
!
!
!
router ospf 4
 log-adjacency-changes
 distance 4
!
!
end
gt3-7200-3# sh run part service

Building configuration...
Current configuration : 190 bytes
!
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
service tcp-small-servers
!
!
!
end
gt3-7200-3# sh run part snmp

Building configuration...
Current configuration : 84 bytes
!
!
!
snmp-server community user101 RW
snmp mib target list qwe host 0.0.0.0
!
end

```

Additional References

The following sections provide references related to the Configuration Partitioning feature.

Related Documents

Related Topic	Document Title
Running configuration performance enhancement-- parserconfigcache for interfaces.	Configuration Generation Performance Enhancement

Related Topic	Document Title
Provisioning of customer services, Config Rollback, Config Locking, and configuration access control	Contextual Configuration Diff Utility
Configuration management--Config change logging.	Configuration Change Notification and Logging
Configuration management --Quick-save for config change logging ¹ .	Configuration Logger Persistency
Cisco IOS software configuration access control and config session locking (“Config Lock”).	Exclusive Configuration Change Access and Access Session Locking

¹ The “Configuration Logger Persistency” feature allows saving just the commands entered since the last startup-config file was generated, rather than saving the entire startup configuration.

Standards

Standard	Title
No standards are associated with this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	--

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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



CHAPTER 4

Configuration Versioning

The Configuration Versioning feature allows you to maintain and manage backup copies of the Cisco running configuration on or off the device. The Configuration Replace feature uses the Configuration Versioning feature to provide a rollback to a saved copy of the running configuration.

- [Finding Feature Information, on page 59](#)
- [Information About Configuration Versioning, on page 59](#)
- [How to Configure Configuration Versioning, on page 60](#)
- [Configuration Examples for Configuration Versioning, on page 64](#)
- [Additional References, on page 64](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Information About Configuration Versioning

Configuration Archive

The Cisco configuration archive provides a mechanism to store, organize, and manage an archive of Cisco configuration files to enhance the configuration rollback capability provided by the **configure replace** command. Before this feature was introduced, you could save copies of the running configuration using the **copy running-config destination-url** command, storing the replacement file either locally or remotely. However, this method lacked any automated file management. With the Configuration Replace and Configuration Rollback feature, you can automatically save copies of the running configuration to the configuration archive. These archived files serve as checkpoint configuration references and can be used by the **configure replace** command to revert the configuration to a previous state.

The **archive config** command allows you to save Cisco configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and

optional time stamp) as each consecutive file is saved. This functionality provides consistent identification of saved configuration files. You can specify how many versions of the running configuration are kept in the archive. After the maximum number of files are saved in the archive, the oldest file is automatically deleted when the next, most recent file is saved. The **show archive** command displays information for all configuration files saved in the configuration archive.

The configuration archive, in which the configuration files are stored and available for use with the **configure replace** command, can be located on the following file systems depending on your platform:

- If your platform has disk0:—disk0:, disk1:, ftp:, pram:, rcpx:, slavedisk0:, slavedisk1:, or tftp:
- If your platform does not have disk0:—bootflash:, ftp:, harddisk:, http:, pram:, rcpx:, tftp:, usb0:, or usb1:

How to Configure Configuration Versioning

Configuring the Characteristics of the Configuration Archive

Before using the **archive config** command, the configuration archive must be configured. Perform this task to configure the characteristics of the configuration archive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **path** *url*
5. **maximum** *number*
6. **time-period** *minutes*
7. **end**
8. **archive config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	archive Example:	Enters archive configuration mode.

	Command or Action	Purpose
	Device(config)# archive	
Step 4	path <i>url</i> Example: Device(config-archive)# path bootflash:myconfig	Specifies the location and filename prefix for the files in the configuration archive. <ul style="list-style-type: none"> Depending on your hardware platform, the name of your file system might be different than shown in the example. Note If a directory is specified in the path instead of file, the directory name must be followed by a forward slash as follows: path flash:/ <i>directory</i> /. The forward slash is not necessary after a filename; it is necessary only when specifying a directory.
Step 5	maximum <i>number</i> Example: Device(config-archive)# maximum 14	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the configuration archive. <ul style="list-style-type: none"> The <i>number</i> argument is the maximum number of archive files of the running configuration to be saved in the configuration archive. The range is 1 to 14. The default is 10. Note Before using this command, you must configure the path command to specify the location and filename prefix for the files in the configuration archive.
Step 6	time-period <i>minutes</i> Example: Device(config-archive)# time-period 10	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the configuration archive. <ul style="list-style-type: none"> The <i>minutes</i> argument specifies how often, in minutes, to automatically save an archive file of the current running configuration in the configuration archive. Note Before using this command, you must configure the path command to specify the location and filename prefix for the files in the configuration archive.
Step 7	end Example: Device(config-archive)# end	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 8	archive config Example: <pre>Device# archive config</pre>	Saves the current running configuration file to the configuration archive. Note You must configure the path command before using the archive config command.

Monitoring and Troubleshooting the Configuration

SUMMARY STEPS

1. enable
2. show archive
3. debug archive versioning
4. debug archive config timestamp
5. exit

DETAILED STEPS

Step 1 enable

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 show archive

Use this command to display information about the files saved in the configuration archive. For example:

Example:

```
Device# show archive

There are currently 1 archive configurations saved.
The next archive file will be named bootflash:myconfig-2
Archive #  Name
0
1      bootflash:myconfig-1 <- Most Recent
2
3
4
5
6
7
8
9
10
11
12
13
14
```

The following is sample output from the **show archive** command after several archive files of the running configuration have been saved. In this example, the maximum number of archive files to be saved is set to three.

Example:

```
Device# show archive
```

```
There are currently 3 archive configurations saved.
```

```
The next archive file will be named bootflash:myconfig-8
```

```
Archive #  Name
0
1          :Deleted
2          :Deleted
3          :Deleted
4          :Deleted
5          bootflash:myconfig-5
6          bootflash:myconfig-6
7          bootflash:myconfig-7 <- Most Recent
8
9
10
11
12
13
14
```

Step 3 debug archive versioning

Use this command to enable debugging of the configuration archive activities to help monitor and troubleshoot configuration replace and rollback. For example:

Example:

```
Device# debug archive versioning
```

```
Jan  9 06:46:28.419:backup_running_config
```

```
Jan  9 06:46:28.419:Current = 7
```

```
Jan  9 06:46:28.443:Writing backup file bootflash:myconfig-7
```

```
Jan  9 06:46:29.547: backup worked
```

Step 4 debug archive config timestamp

Use this command to enable debugging of the processing time for each integral step of a configuration replace operation and the size of the configuration files being handled. For example:

Example:

```
Device# debug archive config timestamp
```

```
Device# configure replace bootflash:myconfig force
```

```
Timing Debug Statistics for IOS Config Replace operation:
```

```
Time to read file slot0:sample_2.cfg = 0 msec (0 sec)
```

```
Number of lines read:55
```

```
Size of file          :1054
```

```
Starting Pass 1
```

```
Time to read file system:running-config = 0 msec (0 sec)
```

```
Number of lines read:93
```

```
Size of file          :2539
```

```
Time taken for positive rollback pass = 320 msec (0 sec)
```

```
Time taken for negative rollback pass = 0 msec (0 sec)
```

```
Time taken for negative incremental diffs pass = 59 msec (0 sec)
```

```
Time taken by PI to apply changes = 0 msec (0 sec)
```

```
Time taken for Pass 1 = 380 msec (0 sec)
```

```
Starting Pass 2
```

```

Time to read file system:running-config = 0 msec (0 sec)
Number of lines read:55
Size of file          :1054
Time taken for positive rollback pass = 0 msec (0 sec)
Time taken for negative rollback pass = 0 msec (0 sec)
Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done

```

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```

Device# exit
Device>

```

Configuration Examples for Configuration Versioning

Example: Creating a Configuration Archive

The following example shows how to perform the initial configuration of the configuration archive. In this example, bootflash:myconfig is specified as the location and filename prefix for the files in the configuration archive and a value of 10 is set as the maximum number of archive files to be saved. Depending on your hardware platform, the name of your file system might be different than shown in the example.

```

configure terminal
!
archive
 path bootflash:myconfig
 maximum 10
end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Information about managing configuration files	“Managing Configuration Files” module in the <i>Managing Configuration Files Configuration Guide</i>
Commands for managing configuration files	Cisco IOS Configuration Fundamentals Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 5

Configuration Rollback Confirmed Change

The Configuration Rollback Confirmed Change feature allows configuration changes to be performed with an optional requirement that they be confirmed. If this confirmation is not received, the configuration is returned to the state prior to the changes being applied. The mechanism provides a safeguard against inadvertent loss of connectivity between a network device and the user or management application due to configuration changes.

- [Finding Feature Information, on page 67](#)
- [Information About Configuration Rollback Confirmed Change, on page 67](#)
- [Restrictions, on page 68](#)
- [How to Configure Configuration Rollback Confirmed Change, on page 68](#)
- [Configuration Examples for Configuration Rollback Confirmed Change, on page 70](#)
- [Additional References, on page 71](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Information About Configuration Rollback Confirmed Change

Configuration Rollback Confirmed Change Operation

The Configuration Rollback Confirmed Change feature enables an added criterion of a confirmation to configuration changes. This functionality enables a rollback to occur if a confirmation of the requested changes is not received in a configured time frame. Command failures can also be configured to trigger a configuration rollback.

The following steps outline how this process is achieved:

1. A new option allows you to request confirmation (a confirmation time limit must be supplied) of the configuration changes.
2. You must enter the confirmation command. If no confirmation is entered within the requested time limit, the configuration reverts to its previous state.

Restrictions

The configure replace command is *not* supported on the T1/E1 interface modules.

How to Configure Configuration Rollback Confirmed Change

Performing a Configuration Replace or Configuration Rollback Operation with Confirmation

Perform this task to replace the current running configuration file with a saved Cisco configuration file.



Note

You must configure a configuration archive before performing this procedure. For detailed steps, see the “Configuring the Characteristics of the Configuration Archive” module in the *Managing Configuration Files Configuration Guide*. The following procedure details how to return to that archived configuration in the event of a problem with the current running configuration.

SUMMARY STEPS

1. **enable**
2. **configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignorecase**] [**revert trigger** [**error**] [**timer minutes**] | **time minutes**]
3. **configure revert** {**now** | **timer** {*minutes* | **idle minutes**}}
4. **configure confirm**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure replace <i>target-url</i> [nolock] [list] [force] [ignorecase] [revert trigger [error] [timer minutes] time minutes]	Replaces the current running configuration file with a saved configuration file.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device# configure replace bootflash:myconfig-1 list time 30</pre>	<ul style="list-style-type: none"> • target-url—Specifies a URL (accessible by the Cisco file system) of the saved configuration file that is to replace the current running configuration, such as the configuration file created by using the archive config command. Depending on your hardware platform, the name of your file system might be different than shown in the example. • no-lock—Disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation. • list—Displays a list of the command lines applied by the Cisco software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed. • force—Replaces the current running configuration file with the specified saved configuration file without prompting you for confirmation. • ignorecase—Allows the configuration to ignore the case of the confirmation command. • time minutes—Specifies the time (in minutes) within which you must enter the configure confirm command to confirm replacement of the current running configuration file. If the configure confirm command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the configure replace command). • revert trigger—Sets the following triggers for reverting to the original configuration: <ul style="list-style-type: none"> • error—Reverts to the original configuration upon error. • timer minutes—Reverts to the original configuration if the specified time elapses.
Step 3	<p>configure revert {now timer {minutes idle minutes}}</p> <p>Example:</p> <pre>Device# configure revert now</pre>	<p>(Optional) Cancels the timed rollback and triggers the rollback immediately or resets parameters for the timed rollback.</p> <ul style="list-style-type: none"> • now—Triggers the rollback immediately. • timer—Resets the configuration revert timer.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Use the <i>minutes</i> argument with the timer keyword to specify a new revert time in minutes. • Use the idle keyword along with a time in minutes to set the maximum allowable time period of no activity before reverting to the saved configuration.
Step 4	configure confirm Example: <pre>Device# configure confirm</pre>	(Optional) Confirms replacement of the current running configuration file with a saved configuration file. Note Use this command only if the time minutes keyword and argument of the configure replace command are specified.
Step 5	exit Example: <pre>Device# exit</pre>	Exits to user EXEC mode.

Configuration Examples for Configuration Rollback Confirmed Change

Example: Performing a Configuration Replace Operation with the **configure confirm** Command

The following example shows the use of the **configure replace** command with the **time minutes** keyword and argument. You must enter the **configure confirm** command within the specified time limit to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the **configure replace** command).

```
Device# configure replace nvram:startup-config time 120
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
Device# configure confirm
```

The following example shows the use of the **configure revert** command with the **timer** keyword. You must enter the **configure revert** command to cancel the timed rollback and trigger the rollback immediately, or to reset parameters for the timed rollback.

```
Device# configure revert timer 100
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Information about managing configuration files	“Managing Configuration Files” module in the <i>Managing Configuration Files Configuration Guide</i>
Commands for managing configuration files	Cisco IOS Configuration Fundamentals Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

