



Managing Configuration Files Configuration Guide, Cisco IOS Release 15M&T

First Published: November 30, 2012

Last Modified: March 29, 2013

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Managing Configuration Files 1

Finding Feature Information 1

Prerequisites for Managing Configuration Files 1

Restrictions for Managing Configuration Files 2

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 3

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 rcp Server 4

Restrictions 4

Requirements for the rcp Username 5

Copy a Configuration File from the Router to an FTP Server 6

Understanding the FTP Username and Password 6

Configuration Files Larger than NVRAM 6

Compressing the Configuration File 6

Storing the Configuration in Flash Memory on Class A Flash File Systems 7

Loading the Configuration Commands from the Network 7

Configure the Router to Download Configuration Files 7

Network Versus Host Configuration Files 7

How to Manage Configuration File Information 8

Displaying Configuration File Information 8

Modifying the Configuration File at the CLI 9

Copying a Configuration File from the Router to a TFTP Server 10

What to Do Next 11

Copying a Configuration File from the Router to an rcp Server 12

Examples	13
Storing a Running Configuration File on an rcp Server	13
Storing a Startup Configuration File on an rcp Server	13
What to Do Next	13
Copying a Configuration File from the Router to the FTP Server	14
Examples	15
Storing a Running Configuration File on an FTP Server	15
Storing a Startup Configuration File on an FTP Server	15
What to Do Next	16
Copying a Configuration File from a TFTP Server to the Router	16
What to Do Next	17
Copying a Configuration File from the rcp Server to the Router	17
Examples	18
Copy rcp Running-Config	18
Copy rcp Startup-Config	19
What to Do Next	19
Copying a Configuration File from an FTP Server to the Router	19
Examples	21
Copy FTP Running-Config	21
Copy FTP Startup-Config	21
What to Do Next	21
Maintaining Configuration Files Larger than NVRAM	21
Compressing the Configuration File	21
Storing the Configuration in Flash Memory on Class A Flash File Systems	23
Loading the Configuration Commands from the Network	25
Copying Configuration Files from Flash Memory to the Startup or Running Configuration	27
Copying Configuration Files Between Flash Memory File Systems	28
Copying a Configuration File from an FTP Server to Flash Memory Devices	29
What to Do Next	31
Copying a Configuration File from an rcp Server to Flash Memory Devices	31
Copying a Configuration File from a TFTP Server to Flash Memory Devices	32
Reexecuting the Configuration Commands in the Startup Configuration File	33
Clearing the Startup Configuration	33
Deleting a Specified Configuration File	34

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems 35

What to Do Next 36

Configuring the Router to Download Configuration Files 37

Configuring the Router to Download the Network Configuration File 37

Configuring the Router to Download the Host Configuration File 39

Additional References 40

CHAPTER 2

Configuration Generation Performance Enhancement 43

Finding Feature Information 43

Restrictions for Configuration Generation Performance Enhancement 43

Information About Configuration Generation Performance Enhancement 44

Cisco IOS Software Configuration Storage 44

Benefits of the Configuration Generation Performance Enhancement 44

How to Configure the Configuration Generation Performance Enhancement 45

Configuring the Configuration Generation Performance Enhancement 45

Configuration Examples for the Configuration Generation Performance Enhancement 46

Configuring the Configuration Generation Performance Enhancement Example 46

Verifying the Configuration Generation Performance Enhancement Example 46

Additional References 46

Feature Information for Configuration Generation Performance Enhancement 48

CHAPTER 3

Exclusive Configuration Change Access and Access Session Locking 49

Finding Feature Information 49

Information About Locking the Configuration 50

Exclusive Configuration Change Access and Access Session Locking 50

Access Session Locking 50

Parser Concurrency and Locking Improvements 50

How to Configure Configuration Exclusive Configuration Change-Access and Access Session Locking 51

Enabling Exclusive Configuration Change Access and Access Session Locking 51

Obtaining Exclusive Configuration Change Access 52

Enabling Parser Concurrency and Locking Improvements 54

Monitoring and Troubleshooting Configuration Locking 55

Configuration Examples for Locking the Configuration 56

Configuring an Exclusive Lock in Auto Mode Example 56

Configuring an Exclusive Lock in Manual Mode Example	56
Configuring Parser Concurrency and Locking Improvements Example	56
Additional References	56
Feature Information for Exclusive Configuration Change Access and Access Session Locking	58

CHAPTER 4

Configuration Replace and Configuration Rollback 63

Finding Feature Information	63
Prerequisites for Configuration Replace and Configuration Rollback	64
Restrictions for Configuration Replace and Configuration Rollback	64
Information About Configuration Replace and Configuration Rollback	65
Configuration Archive	65
Configuration Replace	65
Configuration Rollback	66
Configuration Rollback Confirmed Change Operation	67
Benefits of Configuration Replace and Configuration Rollback	67
How to Use Configuration Replace and Configuration Rollback	67
Creating a Configuration Archive	67
Performing a Configuration Replace or Configuration Rollback Operation	69
Monitoring and Troubleshooting the Feature	71
Configuration Examples for Configuration Replace and Configuration Rollback	74
Creating a Configuration Archive Example	74
Replacing the Current Running Configuration with a Saved Cisco IOS Configuration File Example	74
Reverting to the Startup Configuration File Example	75
Example: Performing a Configuration Replace Operation with the configure confirm Command	75
Performing a Configuration Rollback Operation Example	75
Additional References	76
Feature Information for Configuration Replace and Configuration Rollback	78

CHAPTER 5

Contextual Configuration Diff Utility 81

Finding Feature Information	81
Prerequisites for Contextual Configuration Diff Utility	81
Restrictions for Contextual Configuration Diff Utility	82

Information About Contextual Configuration Diff Utility	82
Benefits of the Contextual Configuration Diff Utility	82
Contextual Configuration Diff Utility Output Format	82
How to Use the Contextual Configuration Diff Utility	83
Performing a Line-by-Line File Comparison Using the Contextual Configuration Diff Utility	83
Configuration Examples for the Contextual Configuration Diff Utility	84
Example: Diff Operation Performed on Running and Startup Configuration Files	84
Example: Incremental Diff Operation Performed on Running and Startup Configuration Files	85
Additional References	86
Feature Information for Contextual Configuration Diff Utility	86

CHAPTER 6

Configuration Change Notification and Logging	89
Finding Feature Information	89
Restrictions for Configuration Change Notification and Logging	90
Information About Configuration Change Notification and Logging	90
Configuration Log	90
Configuration Change Notifications and Config Change Logging	90
How to Configure Configuration Change Notification and Logging	91
Configuring Configuration Change Notification and Logging	91
Displaying Configuration Log Entries and Statistics	92
Clearing Configuration Log Entries	94
Clearing the Configuration Log by Resetting the Log Size	94
Clearing the Configuration Log by Disabling the Configuration Log	96
Configuration Examples for Configuration Change Notification and Logging	97
Example: Configuring Configuration Change Notification and Logging	97
Example: Clearing the Configuration Log by Resetting the Log Size	97
Example: Clearing the Configuration Log by Disabling the Configuration Log	98
Additional References	98
Feature Information for Configuration Change Notification and Logging	98

CHAPTER 7

Configuration Logger Persistency	101
Finding Feature Information	101
Prerequisites for Configuration Logger Persistency	101

Information About Configuration Logger Persistency	102
Use of Configuration Logger Persistency to Save Configuration Files	102
Persisted Commands	102
How to Configure the Configuration Logger Persistency Feature	103
Enabling the Configuration Logger Persistency Feature	103
Verifying and Troubleshooting the Configuration Logger Persistency Feature	105
Configuration Examples for the Configuration Logger Persistency Feature	106
Configuration Logger Persistency Configuration on a Cisco 7200 Series Router	
Example	106
Additional References	106
Feature Information for Configuration Logger Persistency	108
Glossary	108



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, page 1](#)
- [Prerequisites for Managing Configuration Files, page 1](#)
- [Restrictions for Managing Configuration Files, page 2](#)
- [Information About Managing Configuration Files, page 2](#)
- [How to Manage Configuration File Information, page 8](#)
- [Additional References, page 40](#)

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 (see the “Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems” section). The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
 - **nvram:** (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 **copyftp:** **rcp:** **tftp:** **nvr** **ram: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 rcp Server

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

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote shell protocol, which included the remote shell (rsh) and remote copy (rcp) functions. Rsh and rcp give users the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network. The Cisco implementation of rsh and rcp interoperates with standard implementations.

The rcp **copy** commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you need not create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although the Cisco rcp implementation emulates the functions of the UNIX rcp implementation--copying files among systems on the network--the Cisco command syntax differs from the UNIX rcp command syntax. The Cisco rcp support offers a set of **copy** commands that use rcp as the transport mechanism. These rcp **copy** commands are similar in style to the Cisco TFTP **copy** commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the TCP/IP stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You also can enable rcp support to allow users on remote systems to copy files to and from the router.

To configure the Cisco IOS software to allow remote users to copy files to and from the router, use the **ipremdrcp-enable** global configuration command.

Restrictions

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, 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 **iprcmdremote-username** global configuration command, if the command is configured.
- 3 The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
- 4 The router host name.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, you can specify that user name as the remote username.

Use the **iprcmdremote-username** command to specify a username for all copies. (Rcmd is a UNIX routine used at the super-user level to execute commands on a remote machine using an authentication scheme based on reserved port numbers. Rcmd stands for “remote command”). Include the username in the **copy** command if you want to specify a username for that copy operation only.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the rcp server. For example, suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router IP address translates to Router1.company.com, then the .rhosts file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Requirements for the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, 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 **iprcmdremote-username** global configuration command, if the command is configured.
- 3 The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
- 4 The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote 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 rcp server 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 @routename .domain* . The variable *username* is the username associated with the current session, *routename* 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.

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.

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. • 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.)
Step 5	show startup-config Example: Device# show startup-config	Displays the contents of the startup configuration file. (Command alias for the more nvram:startup-config command.) On all platforms except the Class A Flash file system platforms, the default startup-config file usually is stored in NVRAM. On the Class A Flash file system platforms, the CONFIG_FILE environment variable points to the default startup-config file. The CONFIG_FILE variable defaults to NVRAM.

	Command or Action	Purpose
--	-------------------	---------

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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.
Step 3	Do one of the following:	Ends the configuration session and exits to EXEC mode.

	Command or Action	Purpose
	• end • ^Z Example: <pre>Device(config)# end</pre>	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: <pre>Device# copy system:running-config nvram:startup-config</pre>	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.
Router(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

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



Note

Some specific commands might not get saved to NVRAM. You will need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a list of these settings so that you can quickly reconfigure your 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	copy system:running-config tftp: [[[//location]/directory]/filename] Example: Device# copy system:running-config tftp: //server1/topdir/file10	Copies the running configuration file to a TFTP server.
Step 3	copy nvram:startup-config tftp: [[[//location]/directory]/filename] Example: Device# copy nvram:startup-config tftp: //server1/lstdir/file10	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 an rcp Server

To copy a startup configuration file or a running configuration file from the router to an rcp server, use the following commands beginning in privileged EXEC mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username *username***
4. **end**
5. Do one of the following:
 - **copy system:running-config rcp: [[[/[*username@*]*location*]/*directory*]/*filename*]**
 - **copy nvram:startup-config rcp: [[[/[*username@*]*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 rcmd remote-username <i>username</i> Example: Device(config)# ip rcmd remote-username user1	(Optional) Changes the default remote username.
Step 4	end Example: Device(config)# end	(Optional) Exits global configuration mode.
Step 5	Do one of the following: • copy system:running-config rcp: [[[/[<i>username@</i>]<i>location</i>]/<i>directory</i>]/<i>filename</i>]	Specifies that the Device running configuration file be stored on an rcp server. or

	Command or Action	Purpose
	copy nvram:sta rtup-config rcp :[[[/[username@]location]/directory]/filename] Example: <pre>Device# copy system:running-config rcp: //user1@company.com/dir-files/file1</pre>	Specifies that the Device startup configuration file be stored on an rcp server.

Examples

Storing a Running Configuration File on an rcp 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:

```
Router# copy system:running-config rcp://netadmin1@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
Router#
```

Storing a Startup Configuration File on an rcp Server

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

```
Rtr2# configure terminal
Rtr2(config)# ip rcmd remote-username netadmin2
Rtr2(config)# end
Rtr2# copy nvram:startup-config rcp:
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 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: 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	ip ftp username <i>username</i> Example: Router(config)# ip ftp username user1	(Optional) Specifies the default remote username.
Step 4	ip ftp password <i>password</i> Example: Router(config)# ip ftp username guessme	(Optional) Specifies the default password.

	Command or Action	Purpose
Step 5	end Example: Router(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: • copy system:running-config ftp: [[[/[username][:password]@]/location/directory]/filename] • • • copy nvram:startup-config ftp: [[[/[username][:password]@]/location/directory]/filename] Example: Router# copy system:running-config ftp: //user1:guessme@company.com /dir10/file1	Copies the running configuration or startup configuration file to an FTP server.

Examples

Storing a Running Configuration File on an FTP Server

The following example copies the running configuration file named rtr2-confg 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-confg
Write file rtr2-confg 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	copy tftp: [[[//location]/directory]/filename] system:running-config Example: Device# copy tftp://server1/dir10/datasource system:running-config	Copies a configuration file from a TFTP server to the running configuration.
Step 3	copy tftp: [[[//location]/directory]/filename] nvrn:startup-config Example: Device# copy tftp://server1/dir10/datasource nvrn:startup-config	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 the rcp Server to the Router

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

SUMMARY STEPS

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

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	(Optional) Enters configuration mode from the terminal. This step is required only if you override the default remote username (see Step 3).
Step 3	ip rcmd remote-username <i>username</i> Example: Device(config)# ip rcmd remote-username user1	(Optional) Specifies the remote username.

	Command or Action	Purpose
Step 4	end Example: Device(config)# end	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 5	copy rcp: [[[/[username@]location]/directory]/filename]system:running-config Example: Example: or Example: <pre> copy rcp:[[[[/[username@]location]/directory]/filename] nvram:startup-config </pre> Example: <pre> Device# copy rcp://[user1@company.com/dir10/fileone] nvram:s tartup-config </pre>	Copies the configuration file from a rcp server to the running configuration or startup configuration.

Examples

Copy rcp Running-Config

The following example copies a 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@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 rcp from 172.16.101.101

```

Copy rcp 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 rcmd remote-username netadmin1
Rtr2(config)# end
Rtr2# copy rcp: 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 rcp 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.

Copying a Configuration File from an FTP Server to the Router

To copy a configuration file from an FTP 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	(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: 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	copy ftp: [[[/[username [:password]@]location]/directory]/filename]system:running-config Example: Example: or Example: copy ftp: [[[/[username [:password]@]location/directory]/filename] nvram:startup-config Example: Device# copy ftp://user1:guessme@company.com /dir10/datasource nvram:startup-config	Using FTP, copies the configuration file from a network server to running memory or the startup configuration.

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: 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	service compress-config Example: Device(config)# service compress-config	Specifies that the configuration file be compressed.
Step 4	end Example: Device(config)# end	Exits global configuration mode.
Step 5	Do one of the following: • Use FTP, rcp, or TFTP to copy the new configuration. • configure terminal Example: Device# configure terminal	Enters 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 - <i>file-size</i> / <i>buffer-size</i> bytes].”

	Command or Action	Purpose
Step 6	copy system:running-config nvram:startup-config Example: Device(config)# copy system:running-config nvram:startup-config	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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	copy nvram:startup-config flash-filesystem:filename Example: Device# copy nvram:startup-config slot0:Device-config	Copies the current startup configuration to the new location to create the configuration file. Note Slot 0 is not applicable for Cisco ASR 900 RSP3 Module. Use bootflash or USB.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	boot config flash-filesystem: filename Example: Device(config)# boot config slot0:router-config	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 ASR 900 RSP3 Module. Use bootflash or USB.
Step 5	end Example: Device(config)# end	Exits global configuration mode.
Step 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 Example: Device# configure terminal	Enters the new configuration.

	Command or Action	Purpose
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 ASR 900 RSP3 Module. 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	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	Enter your password if prompted.
Step 2	copy system:running-config {ftp: rcp: tftp:} Example: Device# copy system:running-config ftp:	Saves the running configuration to an FTP, rcp, or TFTP server.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
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.

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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • copy filesystem : [partition-number:][filename] nvram:startup-config • copy filesystem : [partition-number:][filename] system:running-config Example: Device# copy slot0:4:ios-upgrade-1 nvram:startup-config	Loads a configuration file directly into NVRAM. or Copies a configuration file to your running configuration. Note Slot 0 is not applicable for Cisco ASR 900 RSP3 Module. 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 ASR 900 RSP3 Module. Use bootflash or USB.

Copying Configuration Files Between Flash Memory File Systems

On platforms with multiple Flash memory file systems, you can copy files from one Flash memory file system, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to another Flash memory file system. Copying files to different Flash memory file systems lets you create backup copies of working configurations and duplicate configurations for other routers. To copy a configuration file between Flash memory file systems, use the following commands in EXEC mode:

SUMMARY STEPS

1. **enable**
2. **show** *source-filesystem* :
3. **copy** *source-filesystem* : [*partition-number:*][*filename*] *dest-filesystem:*[*partition-number:*][*filename*]
4. **verify** *dest-filesystem* : [*partition-number:*][*filename*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show <i>source-filesystem</i> : Example: Device# copy slot0:4:ios-upgrade-1 nvram:startup-config	Displays the layout and contents of Flash memory to verify the filename. Note Slot 0 is not applicable for Cisco ASR 900 RSP3 Module. Use bootflash or USB.
Step 3	copy <i>source-filesystem</i> : [<i>partition-number:</i>][<i>filename</i>] <i>dest-filesystem:</i> [<i>partition-number:</i>][<i>filename</i>] Example: Device# copy flash: slot1:	Copies a configuration file between Flash memory devices. Note Slot 1 is not applicable for Cisco ASR 900 RSP3 Module. Use bootflash or USB. • The source device and the destination device cannot be the same. For example, the copyslot1:slot1: command is invalid.
Step 4	verify <i>dest-filesystem</i> : [<i>partition-number:</i>][<i>filename</i>]	Verifies the checksum of the file you copied. Note Slot 1 is not applicable for Cisco ASR 900 RSP3 Module. Use bootflash or USB.

	Command or Action	Purpose
	Example: Device# verify flash: slot1:	

Example

The following example copies the file named running-config from partition 1 of internal Flash memory to partition 1 of slot 1 on a Cisco 3600 series router. In this example, the source partition is not specified, so the router prompts for the partition number.

```
Device# copy flash: slot1:

System flash
Partition  Size  Used    Free    Bank-Size  State      Copy Mode
1          4096K  3070K   1025K   4096K      Read/Write Direct
2          16384K  1671K   14712K  8192K      Read/Write Direct
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
System flash directory, partition 1:
File Length Name/status
1  3142748 dirt/network/mars-test/c3600-j-mz.latest
2    850   running-config
[3143728 bytes used, 1050576 available, 4194304 total]
PCMCIA Slot1 flash directory:
File Length Name/status
1  1711088 dirt/gate/c3600-i-mz
2    850   running-config
[1712068 bytes used, 2482236 available, 4194304 total]
Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)... OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'running-config' from flash: device
  as 'running-config' into slot1: device WITH erase? [yes/no] yes

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
!
[OK - 850/4194304 bytes]
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum... OK (0x16)
```



Note

Slot 0 or Slot 1 is not applicable for Cisco ASR 900 RSP3 Module. 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. • 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 <i>username</i> Example: Device(config)# ip ftp username user1	(Optional) Specifies the remote username.
Step 4	ip ftp password <i>password</i> 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: [[[/[<i>username:password@</i>] <i>location</i>]/ <i>directory</i>]/ <i>filename</i>] <i>flash-filesystem:[partition-number:]</i> [<i>filename</i>] 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: Device> enable	Enables privileged EXEC mode. • 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 rcmd remote-username <i>username</i> Example: Device(config)# ip rcmd remote-username user1	(Optional) Specifies the remote username.
Step 4	end Example: Device(config)# end	(Optional) Exits configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).

	Command or Action	Purpose
Step 5	copy rcp: [[[/[username@]/location]/directory]/filename] flash-filesystem:[partition-number:]/[filename] Example: Device# copy rcp:router-config slot0:new-config	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 depending 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	copy tftp: [[[/[location]/directory]/filename] flash-filesystem:[partition-number:]/[filename] Example: Device# copy tftp:router-config slot0:new-config	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 depending 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure memory Example: Device# configure memory	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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	erase nvram Example: Device# erase nvram	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 erase startup-config EXEC 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: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	delete flash-filesystem : filename Example: Device# delete slot0:myconfig	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: enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	copy <i>[flash-url ftp-url rcp-url tftp-url system:running-config nvram:startup-config] dest-flash-url</i> Example: Device# copy system:running-config nvram:startup-config	Copies the configuration file to the Flash file system from which the router will load the file upon restart.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	boot config <i>dest-flash-url</i> Example: Device(config)# boot config 172.16.1.1	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.

	Command or Action	Purpose
Step 5	end Example: Device(config)# end	Exits global configuration mode.
Step 6	copy system:running-config nvram:startup-config Example: Device# copy system:running-config nvram:startup-config	Saves the configuration performed in Step 3 to the startup configuration.
Step 7	show boot Example: Device# show boot	(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 **nvram: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 **copy system:running-config nvram: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 Configuration Files

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

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

- "Configuring the Router to Download the Network Configuration File"
- "Configuring the Router to Download the Host Configuration File"

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

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

If there are any problems with the startup configuration file, or if the configuration register is set to ignore NVRAM, the router enters the Setup command facility.

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: 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	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). - 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: 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	boot host {ftp:[[/[/username [:password]@]location]/directory]/filename] rcp:[[/[/username@]location]/directory]/filename] tftp:[[/[/location]/directory]/filename] } Example: Device(config)# boot host tftp:hostfile1	Specifies the host configuration file to download at startup, and the protocol to be used (FTP, rcp, or TFTP): • 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.

	Command or Action	Purpose
Step 4	service config Example: Device(config)# service config	Enables the system to automatically load the host file upon restart.
Step 5	end Example: Device(config)# end	Exits global configuration mode.
Step 6	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.

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 Commands List, All Releases
Cisco IOS configuration commands	<i>Cisco IOS Configuration Fundamentals Command Reference</i>

Standards

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software 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, and support for existing RFCs has not been modified.	--

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

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)



Configuration Generation Performance Enhancement

The Configuration Generation Performance Enhancement feature assists configuration management by enabling faster collection of running configuration file information. This feature is especially useful in managing large networks with numerous interfaces configured.

- [Finding Feature Information, page 43](#)
- [Restrictions for Configuration Generation Performance Enhancement, page 43](#)
- [Information About Configuration Generation Performance Enhancement, page 44](#)
- [How to Configure the Configuration Generation Performance Enhancement, page 45](#)
- [Configuration Examples for the Configuration Generation Performance Enhancement, page 46](#)
- [Additional References, page 46](#)
- [Feature Information for Configuration Generation Performance Enhancement, page 48](#)

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.

Restrictions for Configuration Generation Performance Enhancement

The device on which the Configuration Generation Performance Enhancement feature is used must have enough memory available to store (cache) a large interface configuration file. For example, if the interface

configurations take up 15 KB of memory, using this feature would require having an additional 15 KB of memory space available.

The presence of the **parser config cache interface** command in the module reflects an improvement in the performance of the configured interfaces. There is no such performance improvement observed in modules without any physical interfaces such as the Network Analyzer Module (NAM).

Information About Configuration Generation Performance Enhancement

Cisco IOS Software Configuration Storage

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 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 used by command-line interface (CLI) commands such as **showrunning-config**, **writememory**, and **copysystem:running-configuration** to display or copy the running system configuration. When invoked, NVGEN queries each system component and each instance of interface or other configuration objects. A running configuration file is constructed as NVGEN traverses the system performing these queries.

**Note**

If you try to configure the **writememory** command when a router is low on memory and the backup buffer cannot be allocated, then the command will fail with the error message, "Not enough space." When the **writememory** command fails to apply the new configuration, the backup configuration is used to restore the original configuration.

Benefits of the Configuration Generation Performance Enhancement

Before the Configuration Generation Performance Enhancement feature was introduced, NVGEN always had to query the entire system and could generate only a total configuration. The time required to process the running configuration creates performance problems for configuration management, because completion of the NVGEN operation can take many minutes.

The Configuration Generation Performance Enhancement feature reduces the execution time for NVGEN processes and is especially useful for managing large configuration files that contain numerous interface configurations. This feature provides faster execution of commands that process the running system configuration by caching interface configuration information in system memory, and by retrieving only configuration information that has changed.

How to Configure the Configuration Generation Performance Enhancement

Configuring the Configuration Generation Performance Enhancement

Perform this task to enable the Configuration Generation Performance Enhancement.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parser config cache interface**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <code>Router> enable</code>	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: <code>Router# configure terminal</code>	Enters global configuration mode.
Step 3	parser config cache interface Example: <code>Router(config)# parser config cache interface</code>	Reduces the time required for the CLI to execute commands that manage the running system configuration, especially for large configuration files.
Step 4	end Example: <code>Router(config)# end</code>	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for the Configuration Generation Performance Enhancement

Configuring the Configuration Generation Performance Enhancement Example

The following example shows how to enable the Configuration Generation Performance Enhancement feature:

```
Router(config)# parser config cache interface
```

Verifying the Configuration Generation Performance Enhancement Example

You can verify that the **parserconfigcacheinterface** command has been enabled by checking for the command in the system configuration file displayed when you enter the **showrunning-configuration** EXEC command.



Note

The first time you display the configuration file, you will not see much evidence of improvement in performance because the interface cache will be filled up. However, you will notice performance improvements when you enter subsequent NVGEN-type commands such as the **showrunning-config** EXEC command. Each time the interface configuration changes, the cache of the specified interface is flushed. The other interface data remains cached as before. Entering an NVGEN-type command after modifying the interface configuration will once again not show much evidence of improvement until the next NVGEN-type command is entered.

```
Router# show running-config
!
!
parser config cache interface
!
!
```

Additional References

The following sections provide references related to the Configuration Generation Performance Enhancement feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
System configuration file management commands	The <i>Cisco IOS Configuration Fundamentals Command Reference</i> appropriate to your software release.

Related Topic	Document Title
System configuration file management	“Managing Configuration Files” module in the <i>Cisco IOS Configuration Fundamentals Configuration Guide</i>

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	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

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuration Generation Performance Enhancement

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 1: Feature Information for the Configuration Generation Performance Enhancement Feature

Feature Name	Releases	Feature Information
Configuration Generation Performance Enhancement	12.3(7)T 12.2(25)S 12.2(33)SRC 12.2(33)SB 12.2(33)SXI	The Configuration Generation Performance Enhancement feature assists configuration management by enabling faster collection of running configuration file information. This feature is especially useful in managing large networks with numerous interfaces configured. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. The following command was introduced or modified: parser config cache interface.



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, page 49](#)
- [Information About Locking the Configuration, page 50](#)
- [How to Configure Configuration Exclusive Configuration Change-Access and Access Session Locking, page 51](#)
- [Configuration Examples for Locking the Configuration, page 56](#)
- [Additional References, page 56](#)
- [Feature Information for Exclusive Configuration Change Access and Access Session Locking, page 58](#)

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.

Parser Concurrency and Locking Improvements

In order to overcome the following limitations posed by the Exclusive Configuration Change Access feature, the Parser Concurrency and Locking Improvements feature was introduced in Cisco IOS Release 12.2(33)SRE:

- The Exclusive Configuration Change Access feature locks the configuration to other users. The lock is automatically released when the lock holder exits from the configuration mode. Any other user in the

configuration mode will be returned to the EXEC mode when the lock is acquired. Also, any user can execute the **clearconfigurationlock** command and forcibly remove the lock and allow normal access to all users.

- The router can reload when multiple write processes belonging to the same client simultaneously access the Cisco IOS configurations in a shared mode.
- The router can reload when EXEC commands concurrently access the data structure.

Effective from Cisco IOS Release 12.2(33)SRE, the Concurrency and Locking Improvements feature is the primary locking mechanism used to prevent concurrent configuration of Cisco IOS software by multiple users.

The Parser Concurrency and Locking Improvements feature provides a common interface that ensures that exclusive access is granted to the requested process and prevents others from concurrently accessing the Cisco IOS configuration. It allows access only to the user holding the lock and prevents other clients from accessing the configuration.

Effective from Cisco IOS Release 12.2(33)SRE, the **configurationmodeexclusive {auto | manual}** command will not be available to enable single-user access functionality for the Cisco IOS CLI. Use the **parsercommandserializer** command to enable configuration access only to the users holding the lock and prevent other clients from accessing the configuration.

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	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: <pre>Router> enable</pre>	Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	configuration mode exclusive Example: <pre>Router(config)# configuration mode exclusive</pre>	Enables exclusive configuration change access (configuration lock feature). When the command is enabled, configuration sessions are performed in single-user (exclusive) mode.
Step 4	end Example: <pre>Router(config)# end</pre>	Ends your configuration session and returns the CLI to privileged EXEC mode.

Obtaining Exclusive Configuration Change Access

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	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: <pre>Router> enable</pre>	Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	configure terminal lock Example: <pre>Router(config)# configure terminal lock</pre>	(Optional) Locks the Cisco IOS software in exclusive (single-user) mode. - This command can be used only if you have previously enabled configuration locking by using the configuration mode exclusive command. - 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.	--
Step 5	Do one of the following: end - or exit Example: <pre>Router(config)# end</pre> Example: Example: <pre>Router(config)# exit</pre>	Ends your configuration session, automatically releases the session lock obtained in Step 1, and exits to privileged EXEC mode. 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.

Enabling Parser Concurrency and Locking Improvements

Perform this task to enable configuration access only to the users holding a configuration lock and to prevent other clients from accessing the running configuration.



Note

The Parser Concurrency and Locking Improvements feature does not allow two or more processes to exist simultaneously within the critical section of Cisco IOS configurations.

This feature flags a command to prevent its serialization if an excessive amount of time is required to generate its output or if its use produces more than 10 kilobytes of output. Examples of commands that would not be serialized are the **showterminal** and **showrunning-config** commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parser command serializer**
4. **exit**

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	parser command serializer Example: Router(config)# parser command serializer	Introduces an exclusive lock to serialize access to Cisco IOS configurations.
Step 4	exit Example: Router(config)# exit	(Optional) Exits global configuration mode.

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:

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)#
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

Configuring Parser Concurrency and Locking Improvements Example

The following example shows how to enable the Parser Concurrency and Locking Improvements feature by using the **parsercommandserializer** command:

```
Router# configure terminal
Router(config)# parser command serializer
Router(config)# 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
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Exclusive Configuration Change Access and Access Session Locking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 2: Feature Information for Exclusive Configuration Change Access and Access Session Locking

Feature Name	Releases	Feature Information
Exclusive Configuration Change Access and Access Session Locking	12.3(14)T 12.0(31)S 12.2(33)SRA 12.4(11)T 12.2(33)SXH 12.2(33)SB	

Feature Name	Releases	Feature Information
		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. 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 allowed to run only after the processes initiated by the configuration lock owner have finished. The Exclusive Configuration Change Access feature is complementary with the locking mechanism for the Configuration Replace and Configuration Rollback feature (“rollback lock”). The Configuration Lock feature was integrated into Release 12.0S, and the Access Session Locking feature extension was implemented. The configuration mode exclusive command was extended to include the following keyword options: config_wait, expire, interleave, lock-show, retry_wait, and terminate. The output of the show configuration lock command was improved. The extended feature was integrated into Releases 12.2(33)SRA, 12.4(11)T, 12.2(33)SXH, and 12.2(33)SB. The following sections provide information about this feature: • Information About Locking

Feature Name	Releases	Feature Information
		the Configuration • How to Configure Configuration Lock The following commands were introduced or modified: clear configuration lock, configuration mode exclusive, and configure terminal lock.
Parser Concurrency and Locking Improvements	12.2(33)SRE 15.1(1)T	The Parser Concurrency and Locking Improvements feature provides a common interface that ensures that exclusive access is granted to the requested process and prevents others from concurrently accessing the Cisco IOS configuration. It allows access only to the user holding the lock and prevents other clients from accessing the configuration. The following sections provide information about this feature: • Parser Concurrency and Locking Improvements • Enabling Parser Concurrency and Locking Improvements The following commands were introduced or modified: parser command serializer and test parser session-lock.



Configuration Replace and Configuration Rollback

The Configuration Replace and Configuration Rollback feature provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, effectively rolling back any configuration changes that were made since that configuration file was saved.

- [Finding Feature Information, page 63](#)
- [Prerequisites for Configuration Replace and Configuration Rollback, page 64](#)
- [Restrictions for Configuration Replace and Configuration Rollback, page 64](#)
- [Information About Configuration Replace and Configuration Rollback, page 65](#)
- [How to Use Configuration Replace and Configuration Rollback, page 67](#)
- [Configuration Examples for Configuration Replace and Configuration Rollback, page 74](#)
- [Additional References, page 76](#)
- [Feature Information for Configuration Replace and Configuration Rollback, page 78](#)

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 Configuration Replace and Configuration Rollback

The format of the configuration files used as input by the Configuration Replace and Configuration Rollback feature must comply with standard Cisco software configuration file indentation rules as follows:

- Start all commands on a new line with no indentation, unless the command is within a configuration submode.
- Indent commands within a first-level configuration submode one space.
- Indent commands within a second-level configuration submode two spaces.
- Indent commands within subsequent submodes accordingly.

These indentation rules describe how the software creates configuration files for such commands as **show running-config** or **copy running-config destination-url**. Any configuration file generated on a Cisco device complies with these rules.

Free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration) is required.

Restrictions for Configuration Replace and Configuration Rollback

If the device does not have free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration), the configuration replace operation is not performed.

Certain Cisco configuration commands such as those pertaining to physical components of a networking device (for example, physical interfaces) cannot be added or removed from the running configuration. For example, a configuration replace operation cannot remove the **interface ethernet 0** command line from the current running configuration if that interface is physically present on the device. Similarly, the **interface ethernet 1** command line cannot be added to the running configuration if no such interface is physically present on the device. A configuration replace operation that attempts to perform these types of changes results in error messages indicating that these specific command lines failed.

In very rare cases, certain Cisco configuration commands cannot be removed from the running configuration without reloading the device. A configuration replace operation that attempts to remove this type of command results in error messages indicating that these specific command lines failed.

Information About Configuration Replace and Configuration Rollback

Configuration Archive

The Cisco IOS configuration archive is intended to provide a mechanism to store, organize, and manage an archive of Cisco IOS configuration files to enhance the configuration rollback capability provided by the **configurereplace** command. Before this feature was introduced, you could save copies of the running configuration using the **copyrunning-configdestination-url** command, storing the replacement file either locally or remotely. However, this method lacked any automated file management. On the other hand, the Configuration Replace and Configuration Rollback feature provides the capability to automatically save copies of the running configuration to the Cisco IOS configuration archive. These archived files serve as checkpoint configuration references and can be used by the **configurereplace** command to revert to previous configuration states.

The **archiveconfig** command allows you to save Cisco IOS configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and optional timestamp) as each consecutive file is saved. This functionality provides a means for consistent identification of saved Cisco IOS 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 **showarchive** command displays information for all configuration files saved in the Cisco IOS configuration archive.

The Cisco IOS configuration archive, in which the configuration files are stored and available for use with the **configurereplace** command, can be located on the following file systems:

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

Configuration Replace

The **configurereplace** command provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, effectively rolling back any configuration changes that were made since the previous configuration state was saved.

When using the **configurereplace** command, you must specify a saved Cisco IOS configuration as the replacement configuration file for the current running configuration. The replacement file must be a complete configuration generated by a Cisco IOS device (for example, a configuration generated by the **copyrunning-configdestination-url** command), or, if generated externally, the replacement file must comply with the format of files generated by Cisco IOS devices. When the **configurereplace** command is entered, the current running configuration is compared with the specified replacement configuration and a set of diffs is generated. The algorithm used to compare the two files is the same as that employed by the **showarchiveconfigdifferences** command. The resulting diffs are then applied by the Cisco IOS parser to achieve the replacement configuration state. Only the diffs are applied, avoiding potential service disruption from reapplying configuration commands that already exist in the current running configuration. This algorithm effectively handles configuration changes to order-dependent commands (such as access lists) through a

multiple pass process. Under normal circumstances, no more than three passes are needed to complete a configuration replace operation, and a limit of five passes is performed to preclude any looping behavior.

The Cisco IOS **copysource-urlrunning-config** command is often used to copy a stored Cisco IOS configuration file to the running configuration. When using the **copysource-urlrunning-config** command as an alternative to the **configurereplacetarget-url** command, the following major differences should be noted:

- The **copysource-urlrunning-config** command is a merge operation and preserves all the commands from both the source file and the current running configuration. This command does not remove commands from the current running configuration that are not present in the source file. In contrast, the **configurereplacetarget-url** command removes commands from the current running configuration that are not present in the replacement file and adds commands to the current running configuration that need to be added.
- The **copysource-urlrunning-config** command applies every command in the source file, whether or not the command is already present in the current running configuration. This algorithm is inefficient and, in some cases, can result in service outages. In contrast, the **configurereplacetarget-url** command only applies the commands that need to be applied--no existing commands in the current running configuration are reapplied.
- A partial configuration file may be used as the source file for the **copysource-urlrunning-config** command, whereas a complete Cisco IOS configuration file must be used as the replacement file for the **configurereplacetarget-url** command.

In Cisco IOS Release 12.2(25)S and 12.3(14)T, a locking feature for the configuration replace operation was introduced. When the **configurereplace** command is used, the running configuration file is locked by default for the duration of the configuration replace operation. This locking mechanism prevents other users from changing the running configuration while the replacement operation is taking place, which might otherwise cause the replacement operation to terminate unsuccessfully. You can disable the locking of the running configuration by using the **no**lock keyword when issuing the **configurereplace** command.

The running configuration lock is automatically cleared at the end of the configuration replace operation. You can display any locks that may be currently applied to the running configuration using the **showconfigurationlock** command.



Note

In a scenario when you are performing a configuration replace using a configuration that is not sourced from IOS (such as a custom written configuration) if the login banner has a delimiter that is not the EXT character (ASCII code 003), the banner configuration is rejected and not included in the replaced configuration. Non-working delimiters include ^C, %, #, CC etc.

Configuration Rollback

The concept of rollback comes from the transactional processing model common to database operations. In a database transaction, you might make a set of changes to a given database table. You then must choose whether to commit the changes (apply the changes permanently) or to roll back the changes (discard the changes and revert to the previous state of the table). In this context, rollback means that a journal file containing a log of the changes is discarded, and no changes are applied. The result of the rollback operation is to revert to the previous state, before any changes were applied.

The **configurereplace** command allows you to revert to a previous configuration state, effectively rolling back changes that were made since the previous configuration state was saved. Instead of basing the rollback

operation on a specific set of changes that were applied, the Cisco IOS configuration rollback capability uses the concept of reverting to a specific configuration state based on a saved Cisco IOS configuration file. This concept is similar to the database idea of saving a checkpoint (a saved version of the database) to preserve a specific state.

If the configuration rollback capability is desired, you must save the Cisco IOS running configuration before making any configuration changes. Then, after entering configuration changes, you can use that saved configuration file to roll back the changes (using the **configurereplace target-url** command). Furthermore, since you can specify any saved Cisco IOS configuration file as the replacement configuration, you are not limited to a fixed number of rollbacks, as is the case in some rollback models based on a journal file.

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.

Benefits of Configuration Replace and Configuration Rollback

- Allows you to revert to a previous configuration state, effectively rolling back configuration changes.
- Allows you to replace the current running configuration file with the startup configuration file without having to reload the router or manually undo CLI changes to the running configuration file, therefore reducing system downtime.
- Allows you to revert to any saved Cisco IOS configuration state.
- Simplifies configuration changes by allowing you to apply a complete configuration file to the router, where only the commands that need to be added or removed are affected.
- When using the **configure replace** command as an alternative to the **copy source-url running-config** command, increases efficiency and prevents risk of service outages by not reapplying existing commands in the current running configuration.

How to Use Configuration Replace and Configuration Rollback

Creating a Configuration Archive

No prerequisite configuration is needed to use the **configurereplace** command. Using the **configurereplace** command in conjunction with the Cisco IOS configuration archive and the **archiveconfig** command is optional

but offers significant benefit for configuration rollback scenarios. Before using the **archiveconfig** 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: 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	archive Example: Router(config)# archive	Enters archive configuration mode.
Step 4	path <i>url</i> Example: Router(config-archive)# path disk0:myconfig	Specifies the location and filename prefix for the files in the Cisco IOS configuration archive. 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:/directory/. The forward slash is not necessary after a filename; it is only necessary when specifying a directory.
Step 5	maximum <i>number</i> Example: Router(config-archive)# maximum 14	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the Cisco IOS configuration archive. • The <i>number</i> argument is the maximum number of archive files of the running configuration to be saved in the Cisco IOS configuration archive. Valid values are from 1 to 14. The default is 10.

	Command or Action	Purpose
		Note Before using this command, you must configure the path command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.
Step 6	time-period <i>minutes</i> Example: <pre>Router(config-archive) # time-period 10</pre>	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the Cisco IOS configuration archive. The minutes argument specifies how often, in minutes, to automatically save an archive file of the current running configuration in the Cisco IOS 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 Cisco IOS configuration archive.
Step 7	end Example: <pre>Router(config-archive) # end</pre>	Exits to privileged EXEC mode.
Step 8	archive config Example: <pre>Router# archive config</pre>	Saves the current running configuration file to the configuration archive. Note The path command must be configured before using this command.

Performing a Configuration Replace or Configuration Rollback Operation

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



Note

You must create a configuration archive before performing this procedure. See [Creating a Configuration Archive](#) for detailed steps. 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**] [**reverttrigger**[**error**]/**timer***minutes*]/**timeminutes**]
3. **configure revert** {**now** |**timer**{*minutes*|*idleminutes*}}
4. **configure confirm**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure replace <i>target-url</i> [no lock] [list] [force] [ignorecase] [reverttrigger [error]/ timer <i>minutes</i>] timeminutes Example: <pre>Router# configure replace disk0:myconfig-1 list time 30</pre>	Replaces the current running configuration file with a saved Cisco IOS configuration file. • The <i>target-url</i> argument is a URL (accessible by the Cisco IOS file system) of the saved Cisco IOS configuration file that is to replace the current running configuration, such as the configuration file created using the archiveconfig command. • The list keyword displays a list of the command lines applied by the Cisco IOS software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed. • The force keyword replaces the current running configuration file with the specified saved Cisco IOS configuration file without prompting you for confirmation. • The timeminutes keyword and argument specify the time (in minutes) within which you must enter the configureconfirm command to confirm replacement of the current running configuration file. If the configureconfirm 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 configurereplace command). • The nolock keyword disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation. • The reverttrigger keywords set the following triggers for reverting to the original configuration: • error--Reverts to the original configuration upon error. • timerminutes--Reverts to the original configuration if specified time elapses. • The ignorecase keyword allows the configuration to ignore the case of the confirmation command.

	Command or Action	Purpose
Step 3	configure revert {now timer{minutes idleminutes}} Example: Router# configure revert now Example:	(Optional) To cancel the timed rollback and trigger the rollback immediately, or to reset parameters for the timed rollback, use the configurerevert command in privileged EXEC mode. • now --Triggers the rollback immediately. • timer --Resets the configuration revert timer. • 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: Router# configure confirm	(Optional) Confirms replacement of the current running configuration file with a saved Cisco IOS configuration file. Note Use this command only if the timeseconds keyword and argument of the configurereplace command are specified.
Step 5	exit Example: Router# exit	Exits to user EXEC mode.

Monitoring and Troubleshooting the Feature

Perform this task to monitor and troubleshoot the Configuration Replace and Configuration Rollback feature.

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:

```
Router> enable
Router#
```

Step 2**show archive**

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

Example:

```
Router# show archive
There are currently 1 archive configurations saved.
The next archive file will be named disk0:myconfig-2
Archive #  Name
0
1      disk0:myconfig-1 <- Most Recent
2
3
4
5
6
7
8
9
10
11
12
13
14
```

The following is sample output from the **showarchive** 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:

```
Router# show archive
There are currently 3 archive configurations saved.
The next archive file will be named disk0:myconfig-8
Archive #  Name
0
1      :Deleted
2      :Deleted
3      :Deleted
4      :Deleted
5      disk0:myconfig-5
6      disk0:myconfig-6
7      disk0:myconfig-7 <- Most Recent
8
9
10
11
12
13
14
```

Step 3**debug archive versioning**

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

Example:

```
Router# 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 disk0: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:

```
Router# debug archive config timestamp
Router# configure replace disk0: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:

```
Router# exit
Router>
```

Configuration Examples for Configuration Replace and Configuration Rollback

Creating a Configuration Archive Example

The following example shows how to perform the initial configuration of the Cisco IOS configuration archive. In this example, disk0: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.

```
configure terminal
!
archive
 path disk0:myconfig
 maximum 10
end
```

Replacing the Current Running Configuration with a Saved Cisco IOS Configuration File Example

The following example shows how to replace the current running configuration with a saved Cisco IOS configuration file named disk0:myconfig. The **configure replace** command interactively prompts you to confirm the operation.

```
Device# configure replace disk0:myconfig
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
```

In the following example, the **list** keyword is specified in order to display the command lines that were applied during the configuration replace operation:

```
Device# configure replace disk0:myconfig list
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
!Pass 1
!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro

end
Total number of passes: 1
Rollback Done
```

Reverting to the Startup Configuration File Example

The following example shows how to revert to the Cisco IOS startup configuration file using the **configure replace** command. This example also shows the use of the optional **force** keyword to override the interactive user prompt.

```
Router# configure replace nvram:startup-config force
Total number of passes: 1
Rollback Done
```

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
```

Performing a Configuration Rollback Operation Example

The following example shows how to make changes to the current running configuration and then roll back the changes. As part of the configuration rollback operation, you must save the current running configuration before making changes to the file. In this example, the **archiveconfig** command is used to save the current running configuration. The generated output of the **configure replace** command indicates that only one pass was performed to complete the rollback operation.



Note

Before using the **archiveconfig** command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.

You first save the current running configuration in the configuration archive as follows:

```
archive config
```

You then enter configuration changes as shown in the following example:

```
configure terminal
!
user netops2 password rain
user netops3 password snow
exit
```

After having made changes to the running configuration file, assume you now want to roll back these changes and revert to the configuration that existed before the changes were made. The **showarchive** command is used to verify the version of the configuration to be used as a replacement file. The **configurereplace** command is then used to revert to the replacement configuration file as shown in the following example:

```
Router# show archive
There are currently 1 archive configurations saved.
The next archive file will be named disk0:myconfig-2
Archive #   Name
0
1          disk0:myconfig-1 <- Most Recent
2
3
4
5
6
7
8
9
10
Router# configure replace disk0:myconfig-1
Total number of passes: 1
Rollback Done
```

Additional References

The following sections provide references related to the Configuration Replace and Configuration Rollback feature.

Related Documents

Related Topic	Document Title
Configuration Locking	<i>Exclusive Configuration Change Access and Access Session Locking</i>
Commands for managing configuration files	<i>Cisco IOS Configuration Fundamentals Command Reference</i>
Information about managing configuration files	<i>Managing Configuration Files</i>
Using the Contextual Configuration Diff Utility feature	<i>Contextual Configuration Diff Utility</i>

Standards

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

MIBs

MIBs	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

RFCs	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
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuration Replace and Configuration Rollback

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 3: Feature Information for Configuration Replace and Configuration Rollback

Feature Name	Releases	Feature Information
Configuration Replace and Configuration Rollback		The Configuration Replace and Configuration Rollback feature provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, rolling back any configuration changes that were made since that configuration file was saved. The following sections provide feature information: The following commands were modified by this feature: archive config, configure confirm, configure replace, debug archive config timestamp, debug archive versioning, maximum, path (archive configuration), show archive, show configuration lock, time-period.
Configuration Versioning		The Configuration Versioning feature allows you to maintain and manage backup copies of the Cisco IOS 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.

Feature Name	Releases	Feature Information
Exclusive Configuration Change Access		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. The following command was modified by this feature and applies to the Configuration Replace and Configuration Rollback feature: show configuration lock. Refer to the separate module, Exclusive Configuration Change Access and Access Session Locking, for details
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. This mechanism provides a safeguard against inadvertent loss of connectivity between a network device and the user or management application due to configuration changes. The following sections provide information about this feature: The following commands were modified by this feature: configure confirm,configure replace,configure revert,configure terminal



Contextual Configuration Diff Utility

The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files (accessible through the Cisco Integrated File System [IFS]) and generate a list of the differences between them. The generated output includes information about configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists.

- [Finding Feature Information, page 81](#)
- [Prerequisites for Contextual Configuration Diff Utility, page 81](#)
- [Restrictions for Contextual Configuration Diff Utility, page 82](#)
- [Information About Contextual Configuration Diff Utility, page 82](#)
- [How to Use the Contextual Configuration Diff Utility, page 83](#)
- [Configuration Examples for the Contextual Configuration Diff Utility, page 84](#)
- [Additional References, page 86](#)
- [Feature Information for Contextual Configuration Diff Utility, page 86](#)

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 Contextual Configuration Diff Utility

The format of the configuration files used for the Contextual Configuration Diff Utility feature must comply with standard Cisco configuration file indentation rules as follows:

- Start all commands on a new line with no indentation, unless the command is within a configuration submode.
- Indent commands within a first-level configuration submode one space.
- Indent commands within a second-level configuration submode two spaces.
- Indent commands within subsequent submodes accordingly.

The device must have a contiguous block of memory larger than the combined size of the two configuration files being compared.

Restrictions for Contextual Configuration Diff Utility

If the device does not have a contiguous block of memory larger than the combined size of the two configuration files being compared, the diff operation fails.

Information About Contextual Configuration Diff Utility

Benefits of the Contextual Configuration Diff Utility

The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files (accessible through the Cisco File System [IFS]) and generate a list of the differences between them. The generated output includes information about the following items:

- Configuration lines that have been added, modified, or deleted.
- Configuration modes within which a changed configuration line exists.
- Location changes of configuration lines that are order-sensitive. For example, the **ip access-list** and **community-lists** commands are order-sensitive commands dependent on where they are listed within a configuration file in relation to other commands of similar type.

Contextual Configuration Diff Utility Output Format

Diff Operation

The Contextual Configuration Diff Utility feature uses the filenames of two configuration files as input. A diff operation is performed on the specified files and a list of differences between the two files is generated as output by using the **show archive config differences** command. Interpreting the output is dependent on the order in which the two files are specified in the command. In this section, we assume that the filename of the file entered first is file1 and the filename of the file entered second is file2. Each entry in the generated output list is prefixed with a unique text symbol to indicate the type of difference found. The text symbols and their meanings are as follows:

- A minus symbol (–) indicates that the configuration line exists in file1 but not in file2.
- A plus symbol (+) indicates that the configuration line exists in file2 but not in file1.

- An exclamation point (!) with descriptive comments identifies order-sensitive configuration lines whose location is different in file1 than in file2.

Incremental Diff Operation

Some applications require that the generated output of a diff operation contain configuration lines that are unmodified (in other words, without the minus and plus symbols). For these applications, an incremental diff operation can be performed by using the **show archive config incremental-diffs** command, which compares a specified configuration file to the running configuration file ().

When an incremental diff operation is performed, a list of the configuration lines that do not appear in the running configuration file (in other words, configuration lines that appear only in the specified file that is being compared to the running configuration file) is generated as output. An exclamation point (!) with descriptive comments identifies order-sensitive configuration lines whose location is different in the specified configuration file than in the running configuration file.

How to Use the Contextual Configuration Diff Utility

Performing a Line-by-Line File Comparison Using the Contextual Configuration Diff Utility

SUMMARY STEPS

1. **enable**
2. Enter one of the following:
 - **show archive config differences** *[file1 [file2]]*
 - **show archive config incremental-diffs** *file*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Enter one of the following: • show archive config differences <i>[file1 [file2]]</i> • show archive config incremental-diffs <i>file</i> or	Performs a line-by-line comparison of any two configuration files (accessible through the Cisco IOS File System and generates a list of the differences between them.

	Command or Action	Purpose
	Example: <pre>Device# show archive config differences running-config startup-config</pre> Example: <pre>Device# show archive config incremental-diffs nvram:startup-config</pre>	Performs a line-by-line comparison of a specified configuration file to the running configuration file and generates a list of the configuration lines that do not appear in the running configuration file.
Step 3	exit Example: <pre>Device# exit</pre>	Exits to user EXEC mode.

Configuration Examples for the Contextual Configuration Diff Utility

Example: Diff Operation Performed on Running and Startup Configuration Files

In this example, a diff operation is performed on the running and startup configuration files. The table below shows the configuration files used for this example.

Table 4: Configuration Files Used for the Diff Operation Example

Running Configuration File	Startup Configuration File
<pre>no ip subnet-zero ip cef interface GigabitEthernet1/0/0 ip address 10.7.7.7 255.0.0.0 no ip route-cache no ip mroute-cache duplex half no ip classless snmp-server community public RO</pre>	<pre>ip subnet-zero ip cef ip name-server 10.4.4.4 voice dnis-map 1 dnis 111 interface GigabitEthernet1/0/0 no ip address no ip route-cache no ip mroute-cache shutdown duplex half ip default-gateway 10.5.5.5 ip classless access-list 110 deny ip any host 10.1.1.1 access-list 110 deny ip any host 10.1.1.2 access-list 110 deny ip any host 10.1.1.3 snmp-server community private RW</pre>

The following is sample output from the **show archive config differences** command. This sample output displays the results of the diff operation performed on the configuration files.

```
Device# show archive config differences running-config startup-config

+ip subnet-zero
+ip name-server 10.4.4.4
+voice dnis-map 1
+dnis 111
interface GigabitEthernet1/0/0
+no ip address
+shutdown
+ip default-gateway 10.5.5.5
+ip classless
+access-list 110 deny ip any host 10.1.1.1
+access-list 110 deny ip any host 10.1.1.2
+access-list 110 deny ip any host 10.1.1.3
+snmp-server community private RW
-no ip subnet-zero
interface GigabitEthernet1/0/0
-ip address 10.7.7.7 255.0.0.0
-no ip classless
-snmpp-server community public RO
```

Example: Incremental Diff Operation Performed on Running and Startup Configuration Files

In this example, an incremental diff operation is performed on the startup and running configuration files. The table below shows the configuration files used for this example.

Table 5: Configuration Files Used for the Incremental Diff Operation Example

Startup Configuration File	Running Configuration File
<pre>ip subnet-zero ip cef ip name-server 10.4.4.4 voice dnis-map 1 dnis 111 interface GigabitEthernet1/0/0 no ip address no ip route-cache no ip mroute-cache shutdown duplex half ip default-gateway 10.5.5.5 ip classless access-list 110 deny ip any host 10.1.1.1 access-list 110 deny ip any host 10.1.1.2 access-list 110 deny ip any host 10.1.1.3 snmp-server community private RW</pre>	<pre>no ip subnet-zero ip cef interface GigabitEthernet1/0/0 ip address 10.7.7.7 255.0.0.0 no ip route-cache no ip mroute-cache duplex half no ip classless snmp-server community public RO</pre>

The following is sample output from the **show archive config incremental-diffs** command. This sample output displays the results of the incremental diff operation performed on the configuration files.

```
Device# show archive config incremental-diffs startup-config

ip subnet-zero
ip name-server 10.4.4.4
voice dnis-map 1
```

```

dnis 111
interface GigabitEthernet1/0/0
no ip address
shutdown
ip default-gateway 10.5.5.5
ip classless
access-list 110 deny ip any host 10.1.1.1
access-list 110 deny ip any host 10.1.1.2
access-list 110 deny ip any host 10.1.1.3
snmp-server community private RW

```

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

Feature Information for Contextual Configuration Diff Utility

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/cisco/web/featurenavigator](#). An account on Cisco.com is not required.

Table 6: Feature Information for Contextual Configuration Diff Utility

Feature Name	Releases	Feature Information
Contextual Configuration Diff Utility	12.2(25)S 12.2(27)SBC 12.2(33)SB 12.2(33)SRA 12.2(33)SXH 12.3(4)T 15.0(1)EX Cisco IOS XE Release 2.1	The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files and generate a list of the differences between them. The generated output includes information about configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists. The following commands were introduced or modified: show archive config differences, show archive config incremental-diffs.



Configuration Change Notification and Logging

The Configuration Change Notification and Logging (Config Log Archive) feature allows the tracking of configuration changes entered on a per-session and per-user basis by implementing an archive function. This archive saves *configuration logs* that track each configuration command that is applied, who applied the command, the parser return code (PRC) for the command, and the time the command was applied. This feature also adds a notification mechanism that sends asynchronous notifications to registered applications whenever the configuration log changes.

Before the introduction of the Configuration Change Notification and Logging feature, the only way to determine if the Cisco software configuration had changed was to save a copy of the running and startup configurations to a local computer and do a line-by-line comparison. This comparison method can identify changes that occurred, but does not specify the sequence in which the changes occurred, or the person responsible for the changes.

- [Finding Feature Information, page 89](#)
- [Restrictions for Configuration Change Notification and Logging, page 90](#)
- [Information About Configuration Change Notification and Logging, page 90](#)
- [How to Configure Configuration Change Notification and Logging, page 91](#)
- [Configuration Examples for Configuration Change Notification and Logging, page 97](#)
- [Additional References, page 98](#)
- [Feature Information for Configuration Change Notification and Logging, page 98](#)

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.

Restrictions for Configuration Change Notification and Logging

- Only complete commands input in a configuration mode are logged.
- Commands that are part of a configuration file applied with the **copy** command are not logged.

Information About Configuration Change Notification and Logging

Configuration Log

The Configuration Change Notification and Logging feature tracks changes made to the Cisco software running configuration by maintaining a configuration log. This configuration log tracks changes initiated only through the CLI or HTTP. Only complete commands that result in the invocation of action routines are logged. The following types of entries are not logged:

- Commands that result in a syntax error message
- Partial commands that invoke the device help system

For each configuration command that is executed, the following information is logged:

- The command that was executed
- The configuration mode in which the command was executed
- The name of the user that executed the command
- The time at which the command was executed
- A configuration change sequence number
- Parser return codes for the command

You can display information from the configuration log by using the **show archive log config** command, with the exception of the parser return codes, which are for use by internal Cisco applications only.

Configuration Change Notifications and Config Change Logging

You can configure the Configuration Change and Notification Logging feature to send notification of configuration changes to the software system logging (syslog) process. Syslog notifications allow monitoring of the configuration log information without performing polling and information gathering tasks.

The Configuration Change Notification and Logging feature allows the tracking of configuration changes entered by users on a per-session and per-user basis. This tool allows administrators to track any configuration change made to the software running configuration, and identify the user that made that change.

How to Configure Configuration Change Notification and Logging

Configuring Configuration Change Notification and Logging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **logging enable**
6. **logging size** *entries*
7. **hidekeys**
8. **notify syslog**
9. **end**

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	archive Example: Device(config)# archive	Enters archive configuration mode.
Step 4	log config Example: Device(config-archive)# log config	Enters configuration change logger configuration mode.

	Command or Action	Purpose
Step 5	logging enable Example: <pre>Device(config-archive-log-config)# logging enable</pre>	Enables the logging of configuration changes. Logging of configuration changes is disabled by default.
Step 6	logging size <i>entries</i> Example: <pre>Device(config-archive-log-config)# logging size 200</pre>	(Optional) Specifies the maximum number of entries retained in the configuration log. - Valid values for the <i>entries</i> argument range from 1 to 1000. The default value is 100 entries. - When the configuration log is full, the oldest entry is deleted every time a new entry is added. Note If a new log size is specified that is smaller than the current log size, the oldest log entries are immediately purged until the new log size is satisfied, regardless of the age of the log entries.
Step 7	hidekeys Example: <pre>Device(config-archive-log-config)# hidekeys</pre>	(Optional) Suppresses the display of password information in configuration log files. Note Enabling the hidekeys command increases security by preventing password information from being displayed in configuration log files.
Step 8	notify syslog Example: <pre>Device(config-archive-log-config)# notify syslog</pre>	(Optional) Enables the sending of notifications of configuration changes to a remote syslog.
Step 9	end Example: <pre>Device(config-archive-log-config)# end</pre>	Returns to privileged EXEC mode.

Displaying Configuration Log Entries and Statistics

Perform this task to display entries from the configuration log or statistics about the memory usage of the configuration log. You can enter the commands in any order.

To display configuration log entries and to monitor the memory usage of the configuration log, the Configuration Change Notification and Logging feature provides the **show archive log config** command.

SUMMARY STEPS

1. **enable**
2. **show archive log config** *number* [*end-number*]
3. **show archive log config all provisioning**
4. **show archive log config statistics**
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
```

Step 2

show archive log config *number* [*end-number*]

Use this command to display configuration log entries by record numbers. If you specify a record number for the optional *end-number* argument, all log entries with record numbers in the range from the value entered for the *number* argument through the *end-number* argument are displayed. For example:

```
Device# show archive log config 1 2
```

idx	sess	user@line	Logged command
1	1	user1@console	logging enable
2	1	user1@console	logging size 200

Example:

This example displays configuration log entry numbers 1 and 2. The range for the *number* and *end-number* arguments is 1 to 2147483647.

Step 3

show archive log config all provisioning

Use this command to display all configuration log files as they would appear in a configuration file rather than in tabular format. For example:

Example:

```
Device# show archive log config all provisioning
```

```
archive
log config
logging enable
logging size 200
```

This display also shows the commands used to change configuration modes, which are required to correctly apply the logged commands.

Step 4

show archive log config statistics

Use this command to display memory usage information for the configuration. For example:

Example:

```
Device# show archive log config statistics
```

```
Config Log Session Info:
  Number of sessions being tracked: 1
  Memory being held: 3910 bytes
  Total memory allocated for session tracking: 3910 bytes
  Total memory freed from session tracking: 0 bytes
Config Log log-queue Info:
  Number of entries in the log-queue: 3
  Memory being held in the log-queue: 671 bytes
  Total memory allocated for log entries: 671 bytes
  Total memory freed from log entries: 0 bytes
```

Step 5**exit**

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

Example:

```
Device# exit
Device>
```

Clearing Configuration Log Entries

Entries from the configuration log can be cleared in one of two ways. The size of the configuration log can be reduced by using the **logging size** command, or the configuration log can be disabled and then reenabled with the **logging enable** command.

Clearing the Configuration Log by Resetting the Log Size

This task shows how to clear the configuration log by reducing the log size to 1, then resetting the log size to the desired value, by entering the **logging size** command twice.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **logging size *entries***
6. **logging size *entries***
7. **end**

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	archive Example: Device(config)# archive	Enters archive configuration mode.
Step 4	log config Example: Device(config-archive)# log config	Enters configuration change logger configuration mode.
Step 5	logging size <i>entries</i> Example: Device(config-archive-log-config)# logging size 1	Specifies the maximum number of entries retained in the configuration log. Note Setting the size of the configuration log to 1 results in all but the most recent entry being purged.
Step 6	logging size <i>entries</i> Example: Device(config-archive-log-config)# logging size 200	Specifies the maximum number of entries retained in the configuration log. Note The size of the configuration log should be reset to the desired value after clearing the configuration log.
Step 7	end Example: Device(config-archive-log-config)# end	Exits to privileged EXEC mode.

Clearing the Configuration Log by Disabling the Configuration Log

SUMMARY STEPS

1. enable
2. configure terminal
3. archive
4. log config
5. no logging enable
6. logging enable
7. end

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	archive Example: Device(config)# archive	Enters archive configuration mode.
Step 4	log config Example: Device(config-archive)# log config	Enters configuration change logger configuration mode.
Step 5	no logging enable Example: Device(config-archive-log-config)# no logging enable	Disables the logging of configuration changes. Note Disabling the configuration log results in all records being purged.

	Command or Action	Purpose
Step 6	logging enable Example: Device (config-archive-log-config) # logging enable	Enables the logging of configuration changes.
Step 7	end Example: Device (config-archive-log-config) # end	Exits to privileged EXEC mode.

Configuration Examples for Configuration Change Notification and Logging

Example: Configuring Configuration Change Notification and Logging

The following example shows how to enable configuration logging with a maximum of 200 entries in the configuration log. In the example, security is increased by suppressing the display of password information in configuration log records with the **hidekeys** command, and syslog notifications are turned on with the **notify syslog** command.

```
configure terminal
archive
 log config
 logging enable
 logging size 200
 hidekeys
 notify syslog
```

Example: Clearing the Configuration Log by Resetting the Log Size

The following example shows how to clear the configuration log by reducing the log size to 1, then resetting the log size to the desired value:

```
Device# configure terminal
Device(config)# archive
Device(config-archive)# log config
Device(config-archive-log-config)# logging size 1
Device(config-archive-log-config)# logging size 200
Device(config-archive-log-config)# end
```

Example: Clearing the Configuration Log by Disabling the Configuration Log

The following example clears the configuration log by disabling and then reenabling the configuration log:

```
Device(config)# archive
Device(config-archive)# log config
Device(config-archive-log-config)# no logging enable
Device(config-archive-log-config)# logging enable
Device(config-archive-log-config)# 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

Feature Information for Configuration Change Notification and Logging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [cfn.cisco.com](#). An account on Cisco.com is not required.

Table 7: Feature Information for Configuration Change Notification and Logging

Feature Name	Releases	Feature Information
Configuration Change Notification and Logging		The Configuration Change Notification and Logging (Configuration Logging) feature allows the tracking of configuration changes entered on a per-session and per-user basis by implementing a configuration log. The configuration log tracks each configuration command that is applied, who applied the command, the parser return code for the command, and the time the command was applied. This feature also adds a notification mechanism that sends asynchronous notifications to registered applications whenever the configuration log changes. The following commands were introduced or modified: archive, hidekeys, log config, logging enable, logging size, notify syslog, show archive log config.



Configuration Logger Persistency

The Configuration Logger Persistency feature increases the operational robustness of Cisco IOS configuration and provisioning actions by implementing a “quick-save” functionality. When the Configuration Logger Persistency feature is configured, Cisco IOS software saves just the commands entered since the last startup-config file was generated, rather than saving the entire startup configuration.

- [Finding Feature Information, page 101](#)
- [Prerequisites for Configuration Logger Persistency, page 101](#)
- [Information About Configuration Logger Persistency, page 102](#)
- [How to Configure the Configuration Logger Persistency Feature, page 103](#)
- [Configuration Examples for the Configuration Logger Persistency Feature, page 106](#)
- [Additional References, page 106](#)
- [Feature Information for Configuration Logger Persistency, page 108](#)
- [Glossary, page 108](#)

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 Configuration Logger Persistency

To enable the Configuration Logger Persistency feature, you must have disk0: configured and an external flash card inserted on the router.

To achieve optimum results from the Configuration Logger Persistency feature, you must have Cisco IOS Release 12.2(33)SRA, Release 12.4(11)T, Release 12.2(33)SXH, or Release 12.2(33)SB installed on your system.

Information About Configuration Logger Persistency

Use of Configuration Logger Persistency to Save Configuration Files

Cisco IOS software uses the startup-config file to save router configuration commands across reloads. This single file contains all the commands that need to be applied when the router reboots. The startup-config file gets updated every time a **writememory** command or **copyurl startup-config** command is entered. As the size of the running-config file grows, the time to save the startup-config file to the NVRAM file system increases as well. Startup-config files can be 1 MB and larger. For files of this size, making a single-line change to the startup-config file requires that the entire startup-config file is saved again even though most of the configuration has not changed.

The Configuration Logger Persistency feature implements a “quick-save” functionality. The aim is to provide a “configuration save” mechanism where the time to save changes from the startup-config file is proportional to the size of the incremental changes (with respect to the startup-config file) that need to be saved.

The Cisco IOS configuration logger logs all changes that are manually entered at the command-line prompt. This feature also notifies the registered clients when changes to the log occur. The contents of the configuration log are stored in the run-time memory--the contents of the log are not persisted after reboots.

The Configuration Logger Persistency feature provides a mechanism to persist the configuration commands entered by users across reloads. Only the commands entered at the command-line interface (CLI) (that is, the commands entered in configuration mode) are persisted across reload. This feature uses the Cisco IOS secure file system to persist the configuration commands that are generated.

**Note**

The Cisco IOS configuration logger is different from the system message logging (syslog) facility. Syslog is a general logging facility for tracking system messages. The configuration logger tracks information about configuration commands entered at the CLI.

Persisted Commands

The persisted commands from the Cisco IOS configuration logger are used as an extension to the startup configuration. These saved commands provide a quick-save capability. Rather than saving the entire startup-config file, Cisco IOS software saves just the commands entered since the last startup-config file was generated.

Only the logged commands are persisted. The following additional data from the configuration logger are *not* persisted:

- User who logged the command
- IP address from which the user logged in
- Session and log indexes for the logged command

- Time when the command was entered
- Pre- and post-NVGEN output associated with the entered command
- Parser return code output for the entered command

The persisted commands' primary purpose is for use as a quick-save extension to the startup-config file. The additional information associated with a configuration command is not useful for quick-save purposes. If you need the additional information to be persisted across reboots (for auditing purposes), complete the following steps:

- 1 Enable configuration logger notification to syslog
- 2 Enable the syslog persistence feature

Alternatively, Cisco Networking Services, CiscoView, or other Network Management systems that manage Cisco IOS devices to keep track of configuration changes in an off-the-box storage solution can be used.

By default, upon reload, the persisted commands are appended to the startup-config file. These commands are applied only when you explicitly configure this behavior using a CLI configuration command.

How to Configure the Configuration Logger Persistency Feature

Enabling the Configuration Logger Persistency Feature

The Configuration Logger Persistency feature implements a quick-save mechanism so that the time to save changes from the startup configuration is proportional to the size of the incremental changes (with respect to the startup configuration) that need to be saved. The persisted commands from the Cisco IOS configuration logger will be used as an extension to the startup configuration. The saved commands, which are used as an extension to the startup configuration, provide a quick-save ability. Rather than saving the entire startup-config file, Cisco IOS software saves just the commands entered since the last startup-config file was generated.

To enable the Configuration Logger Persistency feature, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **logging persistent auto manual**
6. **logging persistent reload**
7. **logging size entries**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: <pre>Router> enable</pre>	Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	archive Example: <pre>Router(config)# archive</pre>	Enters archive configuration mode.
Step 4	log config Example: <pre>Router(config-archive)# log config</pre>	Enters archive configuration-log configuration mode.
Step 5	logging persistent auto manual Example: <pre>Router(config-archive-log-cfg)# logging persistent auto</pre>	Enables the Configuration Logger Persistency feature: - The auto keyword specifies that each configuration command will be saved automatically to the Cisco IOS secure file system. - The manual keyword specifies that you can save the configuration commands to the Cisco IOS secure file system on-demand. To do this, you must use the archivelogconfigpersistentsave command. Note To enable the loggingpersistentauto command, you must have disk0: configured and an external flash card inserted on the router.
Step 6	logging persistent reload Example: <pre>Router(config-archive-log-cfg)# logging persistent reload</pre>	Sequentially applies the configuration commands saved in the configuration logger database (since the last writememory command) to the running-config file after a reload.
Step 7	logging size entries Example: <pre>Router(config-archive-log-cfg)# logging size 10</pre>	Specifies the maximum number of entries retained in the configuration log. - Valid values range from 1 to 1000. - The default value is 100 entries.

Verifying and Troubleshooting the Configuration Logger Persistency Feature

Three commands can be used to verify, archive, and clear the contents of the configuration log. For troubleshooting purposes, the command in Step 4 turns on debugging.

SUMMARY STEPS

1. **show archive log config persistent**
2. **clear archive log config persistent**
3. **archive log config persistent save**
4. **debug archive log config persistent**

DETAILED STEPS

Step 1

show archive log config persistent

This command displays the persisted commands in the configuration log. The commands appear in a configlet format. The following is sample output from this command:

Example:

```
Router# show archive log config persistent
!Configuration logger persistentarchive
log config
logging persistent auto
logging persistent reload
archive
log config
logging size 10
logging console
interface loop 101
ip address 10.1.1.1 255.255.255.0
ip address 10.2.2.2 255.255.255.0
no shutdown
```

Step 2

clear archive log config persistent

This command clears the configuration logging persistent database entries. Only the entries in the configuration logging database file are deleted. The file itself is not deleted because it will be used to log new entries. After this command is entered, a message is returned to indicate that the archive log is cleared.

Example:

```
Router# clear archive log config persistent
Purged the config log persist database entries successfully
Router#
```

Step 3

archive log config persistent save

This command saves the configuration log to the Cisco IOS secure file system. For this command to work, the **archivelogconfigpersistentsave** command must be configured.

Step 4

debug archive log config persistent

This command turns on the debugging function. A message is returned to indicate that debugging is turned on.

Example:

```
Router# debug archive log config persistent
debug archive log config persistent debugging is on
```

Configuration Examples for the Configuration Logger Persistency Feature

Configuration Logger Persistency Configuration on a Cisco 7200 Series Router Example

In this example, each configuration command is saved automatically to the Cisco IOS secure file system, configuration commands saved in the configuration logger database (since the last **writememory** command) are applied sequentially to the running-config file, and the maximum number of entries retained in the configuration log is set to 10:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# archive
Router(config-archive)# log config
Router(config-archive-log-config)# logging persistent auto
configuration log persistency feature enabled. Building configuration... [OK]
Router(config-archive-log-config)# logging persistent reload
Router(config-archive-log-config)# logging size 10
Router(config-archive-log-config)# archive log config persistent save
Router(config-archive-log-config)# end
Router#
```

Additional References

The following sections provide references related to the Configuration Logger Persistency feature.

Related Documents

Related Topic	Document Title
Comprehensive command-reference information	<i>Cisco IOS Configuration Fundamentals Command Reference</i>

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported 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.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuration Logger Persistency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

Table 8: Feature Information for Configuration Logger Persistency

Feature Name	Releases	Feature Information
Configuration Logger Persistency	12.2(33)SRA 12.4(11)T 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 3.9S	The Configuration Logger Persistency feature increases the operational robustness of Cisco configuration and provisioning actions by implementing a “quick-save” functionality. Effective with Cisco IOS Release 12.2(33)SRA, Release 12.4(11)T, Release 12.2(33)SXH, and Release 12.2(33)SB, Cisco software saves just the commands entered since the last startup-config file was generated, rather than saving the entire startup configuration. This feature was integrated into Cisco IOS XE Release 3.9S.

Glossary

API --application programming interface.

CAF --command action function.

CDP --Cisco Discovery Protocol.

CSB --Command Status Block.

HA --high-availability architecture.

MIB --Management Information Base.

NAF --NVGEN action function.

NVGEN --nonvolatile generation.

NVRAM --nonvolatile Random Access Memory.

parse chain --A sequence of C language macros defining the syntax of a Cisco IOS command.

RP --Route Processor.

SNMP --Simple Network Management Protocol.

XML --eXtensible Markup Language.

