Managing Configuration Files Configuration Guide, Cisco IOS XE Gibraltar 16.11.x

First Published: 2012-11-30

Last Modified: 2013-03-29

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

© 2019 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Read Me First** 1

CHAPTER 2 **Managing Configuration Files** 3

- Finding Feature Information 3
- Prerequisites for Managing Configuration Files 3
- Restrictions for Managing Configuration Files 3
- Information About Managing Configuration Files 4
  - Types of Configuration Files 4
  - Configuration Mode and Selecting a Configuration Source 4
  - Configuration File Changes Using the CLI 4
  - Location of Configuration Files 5
  - Copy Configuration Files from a Network Server to the Router 5
    - Copy a Configuration File from the Router to a TFTP Server 6
    - Copy a Configuration File from the Router to an FTP Server 6
    - Copying Files Through a VRF 6
  - Configuration Files Larger than NVRAM 7
    - Compressing the Configuration File 7
    - Loading the Configuration Commands from the Network 7
  - Control of the Parser Cache 7
  - Configure the Router to Download Configuration Files 8
    - Network Versus Host Configuration Files 8
- How to Manage Configuration File Information 9
  - Displaying Configuration File Information 9
  - Modifying the Configuration File at the CLI 10
  - Copying a Configuration File from the Router to a TFTP Server 11
    - What to Do Next 12

Copying a Configuration File from the Router to the FTP Server 12

  Examples 13

  What to Do Next 14

Copying a Configuration File from a TFTP Server to the Router 14

  What to Do Next 15

Copying a Configuration File from an FTP Server to the Router 15

  Examples 17

  What to Do Next 17

Maintaining Configuration Files Larger than NVRAM 17

  Compressing the Configuration File 17

Managing the Parser Cache 19

  Clearing the Parser Cache 19

  Disabling the Parser Cache 20

  Reenabling the Parser Cache 20

  What to Do Next 21

Copying Configuration Files from Flash Memory to the Startup or Running Configuration 21

Copying a Configuration File from an FTP Server to Flash Memory Devices 22

  What to Do Next 23

Copying a Configuration File from an rcp Server to Flash Memory Devices 23

Copying a Configuration File from a TFTP Server to Flash Memory Devices 24

Reexecuting the Configuration Commands in the Startup Configuration File 25

Clearing the Startup Configuration 25

Deleting a Specified Configuration File 26

Additional References 27

CHAPTER 3 **Configuration Generation Performance Enhancement** 29

Finding Feature Information 29

Restrictions for Configuration Generation Performance Enhancement 29

Information About Configuration Generation Performance Enhancement 30

  Cisco IOS XE Software Configuration Storage 30

  Benefits of the Configuration Generation Performance Enhancement 30

How to Configure the Configuration Generation Performance Enhancement 30

  Configuring the Configuration Generation Performance Enhancement 30

Configuration Examples for the Configuration Generation Performance Enhancement 31

Configuring the Configuration Generation Performance Enhancement Example **31**
Verifying the Configuration Generation Performance Enhancement Example **31**
Additional References **32**
Feature Information for Configuration Generation Performance Enhancement **33**

CHAPTER 4 **Exclusive Configuration Change Access and Access Session Locking** **35**
Finding Feature Information **35**
Information About Locking the Configuration **36**
Exclusive Configuration Change Access and Access Session Locking **36**
Access Session Locking **36**
How to Configure Configuration Exclusive Confguration Change-Access and Access Session Locking **37**
Enabling Exclusive Configuration Change Access and Access Session Locking **37**
Obtaining Exclusive Configuration Change Access **38**
Monitoring and Troubleshooting Configuration Locking **39**
Configuration Examples for Locking the Configuration **40**
Configuring an Exclusive Lock in Auto Mode Example **40**
Configuring an Exclusive Lock in Manual Mode Example **40**
Additional References **40**
Feature Information for Exclusive Configuration Change Access and Access Session Locking **41**

CHAPTER 5 **Configuration Replace and Configuration Rollback** **43**
Finding Feature Information **43**
Prerequisites for Configuration Replace and Configuration Rollback **43**
Restrictions for Configuration Replace and Configuration Rollback **44**
Information About Configuration Replace and Configuration Rollback **44**
Configuration Archive **44**
Configuration Replace **45**
Configuration Rollback **46**
Configuration Rollback Confirmed Change Operation **46**
Benefits of Configuration Replace and Configuration Rollback **47**
How to Use Configuration Replace and Configuration Rollback **47**
Creating a Configuration Archive **47**
Performing a Configuration Replace or Configuration Rollback Operation **49**

Monitoring and Troubleshooting the Feature 51
Configuration Examples for Configuration Replace and Configuration Rollback 53
Creating a Configuration Archive Example 53
Replacing the Current Running Configuration with a Saved Cisco IOS Configuration File Example 54
Reverting to the Startup Configuration File Example 54
Example: Performing a Configuration Replace Operation with the configure confirm Command 54
Performing a Configuration Rollback Operation Example 55
Additional References 56
Feature Information for Configuration Replace and Configuration Rollback 57

CHAPTER 6 Contextual Configuration Diff Utility 59
Finding Feature Information 59
Prerequisites for Contextual Configuration Diff Utility 59
Restrictions for Contextual Configuration Diff Utility 60
Information About Contextual Configuration Diff Utility 60
Benefits of the Contextual Configuration Diff Utility 60
Contextual Configuration Diff Utility Output Format 60
How to Use the Contextual Configuration Diff Utility 61
Performing a Line-by-Line File Comparison Using the Contextual Configuration Diff Utility 61
Configuration Examples for the Contextual Configuration Diff Utility 62
Diff Operation Example 62
Incremental Diff Operation Example 64
Additional References 66
Feature Information for Contextual Configuration Diff Utility 67

CHAPTER 7 Configuration Change Notification and Logging 69
Finding Feature Information 69
Restrictions for Configuration Change Notification and Logging 69
Information About Configuration Change Notification and Logging 70
Configuration Log 70
Configuration Change Notifications and Config Change Logging 70
Config Logger Enhancements for EAL4+ Certification 70

How to Configure Configuration Change Notification and Logging 71
  Configuring Configuration Change Notification and Logging 71
  Displaying Configuration Log Entries and Statistics 72
  Clearing Configuration Log Entries 74
    Clearing the Configuration Log by Resetting the Log Size 74
    Clearing the Configuration Log by Disabling the Configuration Log 75
Configuration Examples for Configuration Change Notification and Logging 77
  Example: Configuring Configuration Change Notification and Logging 77
Additional References 77
Feature Information for Configuration Change Notification and Logging 78

CHAPTER 8 **Configuration Partitioning** 79
Finding Feature Information 79
Information About Configuration Partitioning 80
  System Running Configurations 80
  Retrieving the Running Configuration for Display or Copy Operations 80
  Benefits of Partitioning the Running Configuration 80
How to Use the Configuration Partitioning Feature 81
  Displaying Configuration Partitions 81
  Disabling the Configuration Partitioning Feature 83
    What to Do Next 83
Configuration Examples for Configuration Partitioning 84
  Displaying Configuration Partitions Example 84
Additional References 93
Feature Information for Configuration Partitioning 95

CHAPTER 9 **Configuration Versioning** 97
Finding Feature Information 97
Information About Configuration Versioning 97
  Configuration Archive 97
How to Configure Configuration Versioning 98
  Configuring the Characteristics of the Configuration Archive 98
  Monitoring and Troubleshooting the Configuration 100
Configuration Examples for Configuration Versioning 102

Example: Creating a Configuration Archive 102
Additional References 102
Feature Information for Configuration Versioning 103

CHAPTER 10 Configuration Rollback Confirmed Change 105
Finding Feature Information 105
Information About Configuration Rollback Confirmed Change 105
Configuration Rollback Confirmed Change Operation 105
How to Configure Configuration Rollback Confirmed Change 106
Performing a Configuration Replace or Configuration Rollback Operation with Confirmation 106
Configuration Examples for Configuration Rollback Confirmed Change 108
Example: Performing a Configuration Replace Operation with the configure confirm Command 108
Additional References 109
Feature Information for Configuration Rollback Confirmed Change 109

CHAPTER 11 Configuration Logger Persistency 111
Finding Feature Information 111
Prerequisites for Configuration Logger Persistency 111
Information About Configuration Logger Persistency 112
Use of Configuration Logger Persistency to Save Configuration Files 112
Persisted Commands 112
How to Configure the Configuration Logger Persistency Feature 113
Enabling the Configuration Logger Persistency Feature 113
Verifying and Troubleshooting the Configuration Logger Persistency Feature 115
Configuration Examples for the Configuration Logger Persistency Feature 116
Configuration Logger Persistency Configuration on a Cisco 7200 Series Router Example 116
Additional References 116
Feature Information for Configuration Logger Persistency 117
Glossary 118

CHAPTER 12 Software Maintenance Upgrade 119
Information About Software Maintenance Upgrade 119
Software Maintenance Upgrade 119

  - Supported Platforms **120**
  - Software Maintenance Upgrade Package **120**
  - Software Maintenance Upgrade Workflow **120**
  - SMU Reload **120**
- Restrictions for Software Maintenance Upgrade **120**
- How to Configure Software Maintenance Upgrade **121**
  - Adding, Activating, Committing an SMU **121**
  - Rolling Back, Deactivating, or Removing an SMU **121**
- Configuration Examples for Software Maintenance Upgrade **122**
  - Example: Adding, Activating, and Committing an SMU **122**
  - Example: Rolling Back, Deactivating, or Removing an SMU **124**
- Additional References for Software Maintenance Upgrade **128**
- Feature Information for Software Maintenance Upgrade **128**

CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

CHAPTER 2

Managing Configuration Files

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

- Finding Feature Information, on page 3
- Prerequisites for Managing Configuration Files, on page 3
- Restrictions for Managing Configuration Files, on page 3
- Information About Managing Configuration Files, on page 4
- How to Manage Configuration File Information, on page 9
- Additional References, on page 27

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Managing Configuration Files

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.
- You should have at least a minimal configuration running on your system. You can create a basic configuration file using the **setup** command (see Using Setup Mode to Configure a Cisco Networking Device for details).

Restrictions for Managing Configuration Files

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

Information About Managing Configuration Files

Types of Configuration Files

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

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

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

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

Configuration Mode and Selecting a Configuration Source

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

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

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

Configuration File Changes Using the CLI

The Cisco IOS software accepts one configuration command per line. You can enter as many configuration commands as you want. You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the**show running-config or more system:running-config** EXEC command. Comments do not display when you list the startup configuration with the **show startup-config** or **more nvram:startup-config** EXEC mode command. Comments are stripped out of the configuration file when it is loaded onto the router. However, you can list the comments in configuration files stored on a File Transfer Protocol (FTP), remote copy protocol (rcp), or Trivial File Transfer Protocol (TFTP) server. When you configure the software using the CLI, the software executes the commands as you enter them.

Location of Configuration Files

Configuration files are stored in the following locations:

- The running configuration is stored in RAM.
- On all platforms except the Class A Flash file system platforms, the startup configuration is stored in nonvolatile random-access memory (NVRAM).
- On Class A Flash file system platforms, the startup configuration is stored in the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
    - **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**EXECcommand 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:**} **nvram:startup-config**command) and reload the router.

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

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

Copy a Configuration File from the Router to a TFTP Server

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

Copy a Configuration File from the Router to an FTP Server

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

Understanding the FTP Username and Password

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

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

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

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

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

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

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

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

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

Copying Files Through a VRF

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

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

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

Configuration Files Larger than NVRAM

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

Compressing the Configuration File

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

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

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

The **servicecompress-config**global configurationcommand 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
```

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 configura tions, see the "Copying a Configuration File from the Router to a TFTP Server" and "Configure the Router to Download Configuration Files" sections for more information on these commands.

Control of the Parser Cache

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

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

The Parser Cache is enabled by default on all platforms using Cisco IOS Release 12.1(5)T and later releases. However, users with Cisco devices that do not require large configuration files may want to disable the Parser Cache to free the resources used by this feature. (Memory used by this feature depends on the size of the configuration files parsed, but is generally less than 512 KB.)

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

- Clearing the Parser Cache--To free resources or to reset the parser cache memory, you may wish to clear the parse entries and hit/miss statistics stored by the Parser Cache feature
- Disabling the Parser Cache--The Parser Cache feature is enabled by default. To disable the Parser Cache feature, use the no parser cache command in global configuration mode. When the parser cache is disabled, the **noparsercache** command line is written to the running configuration file. If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **noparsercache** command. You will not be able to clear the parser cache after disabling it.
- Reenabling the Parser Cache--To reenable the Parser Cache feature after disabling it, use the **parsercache** command in global configuration mode
- Monitoring the Parser--Statistics about the last configuration file parsed are kept in the system memory, along with hit/miss statistics on the commands parsed by the Parser Cache feature. “Hits” and “misses” refer to the matches that the parser cache was able to make to similar commands used previously in the configuration session. Those commands that are matched (“hits”) be parsed more efficiently. The parser cache cannot improve the parse time for those commands it was unable to match (“misses”).

Configure the Router to Download Configuration Files

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

Network Versus Host Configuration Files

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

How to Manage Configuration File Information

Displaying Configuration File Information

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • 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** *file-url* **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.

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 **showstartup-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`	Enters global configuration mode. Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.
Step 3	Do one of the following: • **end** • **^Z** **Example:** `Device(config)# end`	Ends the configuration session and exits to EXEC mode. **Note** When you press the Ctrl and Z keys simultaneously, ^Z is displayed to the screen.
Step 4	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the running configuration file as the startup configuration file. You may also use the **copy running-config startup-config** command alias, but you should be aware that this command

	Command or Action	Purpose
		is less precise. On most platforms, this command saves the configuration to NVRAM. On the Class A Flash file system platforms, this step saves the configuration to the location specified by the CONFIG_FILE environment variable (the default CONFIG_FILE variable specifies that the file should be saved to NVRAM).

Examples

In the following example, the Device prompt name of the Device is configured. The comment line, indicated by the exclamation mark (!), does not execute any command. The **hostname** command is used to change the Device name from Device to new_name. By pressing Ctrl-Z (^Z) or entering the **end**command, the user quits configuration mode. The **copy system:running-config nvram:startup-config** command saves the current configuration to the startup configuration.

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

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

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

Copying a Configuration File from the Router to a TFTP Server

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
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: //server1/1stdir/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-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] y
Writing tokyo-confg!!! [OK]
```

What to Do Next

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

Copying a Configuration File from the Router to the FTP Server

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

SUMMARY STEPS

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

- **copy nvram:sta rtup-config ftp:** [[[//[*username* [**:***password*]**@**]*location*/*directory*]/*filename*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ftp username** *username* **Example:** `Device(config)# ip ftp username user1`	(Optional) Specifies the default remote username.
Step 4	**ip ftp password** *password* **Example:** `Device(config)# ip ftp username guessme`	(Optional) Specifies the default password.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 6	Do one of the following: • **copy system:running-config ftp:** [[[//[*username* [**:***password*]**@**]*location*/*directory*]/*filename*] • • • **copy nvram:sta rtup-config ftp:** [[[//[*username* [**:***password*]**@**]*location*/*directory*]/*filename*] **Example:** `Device# 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-confg]?
Write file rtr2-confg 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*] **nvram:startup-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	**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*] **nvram:startup-config** **Example:** `Device# copy tftp://server1/dir10/datasource nvram: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-confg system:running-config

Configure using tokyo-confg from 172.16.2.155? [confirm] y

Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

What to Do Next

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

Copying a Configuration File from an FTP Server to the Router

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
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** *username* **Example:** `Device(config)# ip ftp username user1`	(Optional) Specifies the default remote username.
Step 4	**ip ftp password** *password* **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-confg 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-confg system:running-config

Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![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-confg 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-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![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 - *file-size* /*buffer-size* bytes].”
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-confg system:running-config

Configure using tokyo-confg from 172.16.2.155? [confirm] y

Booting tokyo-confg 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]
```

Managing the Parser Cache

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

Clearing the Parser Cache

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

SUMMARY STEPS

1. **enable**
2. **clear parser cache**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**clear parser cache** **Example:** `Device# clear parser cache`	Clears the parse cache entries and hit/miss statistics stored for the Parser Cache feature.

Disabling the Parser Cache

The Parser Cache feature is enabled by default. To disable the Parser Cache feature, complete the task in this section:

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no parser cache** **Example:** `Device(config)# no parser cache`	Disables the Parser Cache feature: • When the parser cache is disabled, the **noparsercache** command line is written to the running configuration file. • If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **noparsercache** command. You will not be able to clear the parser cache after disabling it.

Reenabling the Parser Cache

To reenable the Parser Cache feature after disabling it, complete the task in this section:

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	- Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parser cache** **Example:** `Device(config)# parser cache`	Enables the Parser Cache feature.

What to Do Next

The**showparserstatistics** command displays two sets of data, as follows:

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

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

To copy a configuration file from Flash memory directly to your startup configuration in NVRAM or your running configuration, enter one of the commands in Step 2:

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.

	Command or Action	Purpose
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.

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]
```

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.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 3	**ip ftp username** *username* **Example:** `Device(config)# ip ftp username user1`	(Optional) Specifies the remote username.
Step 4	**ip ftp password** *password* **Example:** `Device(config)# ip ftp password guessme`	(Optional) Specifies the remote password.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits configuration mode. This step is required only if you override the default remote username (see Steps 3 and 4).
Step 6	**copy ftp:** [[[//[*username*:*password*@]*location*]/*directory*]/*filename*] *flash-filesystem*:[*partition-number*:][*filename*] **Example:** `Device> copy ftp:router-config slot0:new-config`	Copies the configuration file from a network server to the Flash memory device using FTP.

What to Do Next

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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** *username* **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).
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.

	Command or Action	Purpose
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.
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 **erasestartup-config**EXECcommand, 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 specifi ed 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.

	Command or Action	Purpose
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.

Additional References

Related Documents

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

Technical Assistance

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

CHAPTER 3

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, on page 29
- Restrictions for Configuration Generation Performance Enhancement, on page 29
- Information About Configuration Generation Performance Enhancement, on page 30
- How to Configure the Configuration Generation Performance Enhancement, on page 30
- Configuration Examples for the Configuration Generation Performance Enhancement, on page 31
- Additional References, on page 32
- Feature Information for Configuration Generation Performance Enhancement, on page 33

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

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.

Information About Configuration Generation Performance Enhancement

Cisco IOS XE Software Configuration Storage

In the Cisco IOS XE 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 **show running-configuration**, **write memory**, and **copy system: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.

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:** `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	**parser config cache interface** **Example:** `Device(config)# parser config cache interface`	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:** `Device(config)# end`	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:

```
Device(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.

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

Additional References

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

Related Documents

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

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

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
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 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 www.cisco.com/go/cfn. 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		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. Commands associated with this feature: • **parser config cache interface** • **parser config partition** • **parser cache**

CHAPTER 4

Exclusive Configuration Change Access and Access Session Locking

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

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

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

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About Locking the Configuration

Exclusive Configuration Change Access and Access Session Locking

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

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

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

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

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

Access Session Locking

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

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

Enabling Exclusive Configuration Change Access and Access Session Locking

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**configuration mode exclusive** **Example:** `Router(config)# configuration mode exclusive`	Enables exclusive configuration change access (configuration lock feature). • When the command is enabled, configuration sessions are performed in single-user (exclusive) mode.
Step 4	**end** **Example:** `Router(config)# end`	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** **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	**configure terminal lock** **Example:** `Router(config)# configure terminal lock`	(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:** `Router(config)# end` **Example:**	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.

	Command or Action	Purpose
	Example: `Router(config)# exit`	

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

Additional References

The following sections provide references related to locking the configuration.

Related Documents

Related Topic	Document Title
Commands for managing configuration files	*Cisco IOS Configuration Management Command Reference*
Information about managing configuration files	*Managing Configuration Files*

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 www.cisco.com/go/cfn. 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	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 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 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**.

CHAPTER 5

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, on page 43
- Prerequisites for Configuration Replace and Configuration Rollback, on page 43
- Restrictions for Configuration Replace and Configuration Rollback, on page 44
- Information About Configuration Replace and Configuration Rollback, on page 44
- How to Use Configuration Replace and Configuration Rollback, on page 47
- Configuration Examples for Configuration Replace and Configuration Rollback, on page 53
- Additional References, on page 56
- Feature Information for Configuration Replace and Configuration Rollback, on page 57

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

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-config***destination-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:, rcp:, slavedisk0:, slavedisk1:, or tftp:
- If your platform does not have disk0--ftp:, http:, pram:, rcp:, 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-config***destination-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 **copy***source-url***running-config** command is often used to copy a stored Cisco IOS configuration file to the running configuration. When using the **copy***source-url***running-config**command as an alternative to the **configurereplace***target-url*command, the following major differences should be noted:

- The **copy***source-url***running-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 **configurereplace***target-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 **copy***source-url***running-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 **configurereplace***target-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 **copy***source-url***running-config** command, whereas a complete Cisco IOS configuration file must be used as the replacement file for the **configurereplace***target-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 **nolock** 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:** `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	**path** *url* **Example:** `Device(config-archive)# path flash: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** *number* **Example:** `Device(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 *number* 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. **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** *minutes* **Example:** `Device(config-archive)# time-period 10`	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the 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.

	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 7	**end** **Example:** `Device(config-archive)# end`	Exits to privileged EXEC mode.
Step 8	**archive config** **Example:** `Device# archive config`	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*]|**time***minutes*]
3. **configure revert** {**now** |**timer**{*minutes*|**idle***minutes*}}
4. **configure confirm**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignorecase**] [**reverttrigger[error]**/**timer***minutes*]|**time***minutes*]	Replaces the current running configuration file with a saved Cisco IOS configuration file.

	Command or Action	Purpose

	Example:
	`Device# configure replace flash:myconfig-1 list time 30`
		- The *target -url* 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 **time***minutes*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.
		  - **timer***minutes*--Reverts to the original configuration if specified time elapses.
		- The **ignorecase** keyword allows the configuration to ignore the case of the confirmation command.

Step 3	**configure revert {now |timer{***minutes***|idle***minutes***}}** **Example:** `Device# 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.

	Command or Action	Purpose
		- Use the *minutes* 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:** `Device# 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 **time***seconds* keyword and argument of the **configurereplace** command are specified.
Step 5	**exit** **Example:** `Device# 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:

```
Device> enable
Device#
```

Step 2 **show archive**

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

Example:

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfig-2
 Archive #  Name
   0
   1       flash: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:

```
Device# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:myconfig-8
 Archive #  Name
   0
   1       :Deleted
   2       :Deleted
   3       :Deleted
   4       :Deleted
   5       flash:myconfig-5
   6       flash:myconfig-6
   7       flash: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:

```
Device# debug archive versioning
Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file flash:myconfig-7
Jan  9 06:46:29.547: backup worked
```

Step 4 **debug archive config timestamp**

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

Example:

```
Device# debug archive config timestamp
Device# configure replace flash:myconfig force
Timing Debug Statistics for IOS Config Replace operation:
       Time to read file slot0:sample_2.cfg = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
Starting Pass 1
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:93
       Size of file        :2539
       Time taken for positive rollback pass = 320 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for negative incremental diffs pass = 59 msec (0 sec)
       Time taken by PI to apply changes = 0 msec (0 sec)
       Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
       Time taken for positive rollback pass = 0 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done
```

Step 5 **exit**

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

Example:

```
Device# exit
Device>
```

Configuration Examples for Configuration 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, flash: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 flash: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 flash:myconfig. The **configure replace** command interactively prompts you to confirm the operation.

```
Device# configure replace flash: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 flash: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 **configurereplace** command. This example also shows the use of the optional **force** keyword to override the interactive user prompt.

```
Device# 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 **configurereplace** 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:

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfig-2
 Archive #  Name
   0
   1       flash:myconfig-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
Device# configure replace flash: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	*Exclusive Configuration Change Access and Access Session Locking*
Commands for managing configuration files	*Cisco IOS Configuration Fundamentals Command Reference*
Information about managing configuration files	*Managing Configuration Files*
Using the Contextual Configuration Diff Utility feature	*Contextual Configuration Diff Utility*

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 www.cisco.com/go/cfn. 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**

CHAPTER 6

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 IOS XE Integrated File System [IFS]) and generate a list of the differences between them. The generated output includes information regarding configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists.

- Finding Feature Information, on page 59
- Prerequisites for Contextual Configuration Diff Utility, on page 59
- Restrictions for Contextual Configuration Diff Utility, on page 60
- Information About Contextual Configuration Diff Utility, on page 60
- How to Use the Contextual Configuration Diff Utility, on page 61
- Configuration Examples for the Contextual Configuration Diff Utility, on page 62
- Additional References, on page 66
- Feature Information for Contextual Configuration Diff Utility, on page 67

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 IOS XE 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 router 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 IOS XE File System [IFS]) and generate a list of the differences between them. The generated output includes information regarding 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 Cisco IOS XE 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** [*file1* [*file2*]]
	• **show archive config incremental-diffs** *file*
	Example:
	`Device# show archive config differences running-config startup-config`
	Example:
	`Device# show archive config incremental-diffs nvram:startup-config`
		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.
		or
		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.

	Command or Action	Purpose
Step 3	**exit** **Example:** `Device# exit`	Exits to user EXEC mode.

Configuration Examples for the Contextual Configuration Diff Utility

Diff Operation Example

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

```
no ip subnet-zero

ip cef

interface FastEthernet1/0

 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
```

Startup Configuration File

```
ip subnet-zero

ip cef

ip name-server 10.4.4.4

voice dnis-map 1

 dnis 111

interface FastEthernet1/0

 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
```

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 in the table below.

```
Device# show archive config differences system:running-config nvram:startup-config

+ip subnet-zero

+ip name-server 10.4.4.4

+voice dnis-map 1
```

```
 +dnis 111

interface FastEthernet1/0

 +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 FastEthernet1/0

 -ip address 10.7.7.7 255.0.0.0

-no ip classless

-snmp-server community public RO
```

Incremental Diff Operation Example

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

```
ip subnet-zero
ip cef
ip name-server 10.4.4.4
voice dnis-map 1
 dnis 111
interface FastEthernet1/0
 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
```

Running Configuration File

```
no ip subnet-zero
ip cef
interface FastEthernet1/0
 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
```

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 in the table below.

```
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 FastEthernet1/0

 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

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

Related Documents

Related Topic	Document Title
Running configuration performance enhancement--**parserconfigcache** for interfaces.	Configuration Generation Performance Enhancement
Provisioning of customer services, Config Rollback, Config Locking, and configuration access control	Contextual Configuration Diff Utility
Configuration management--Config change logging.	Configuration Change Notification and Logging
Configuration management --Quick-save for config change logging[2].	Configuration Logger Persistency
Cisco IOS software configuration access control and config session locking (“Config Lock”).	Exclusive Configuration Change Access and Access Session Locking

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

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
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 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 www.cisco.com/go/cfn. 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	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 regarding configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following commands were modified by this feature: **show archive config differences**, **show archive config incremental-diffs**.

CHAPTER 7

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, on page 69
- Restrictions for Configuration Change Notification and Logging, on page 69
- Information About Configuration Change Notification and Logging, on page 70
- How to Configure Configuration Change Notification and Logging, on page 71
- Configuration Examples for Configuration Change Notification and Logging, on page 77
- Additional References, on page 77
- Feature Information for Configuration Change Notification and Logging, on 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.

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.

Config Logger Enhancements for EAL4+ Certification

The Config Logger Enhancements for EAL4+ Certification feature ensures that the logging process meets the requirements set forth in the Conformance to Common Criteria, Evaluation Assurance Level 4+ (EAL4+) Firewall Protection Profiles. These enhancements include changes to meet the following requirements:

- If you change any logging parameters, those changes are logged. This is effected by the sending of a syslog message for each change to the running configuration from a copy operation (for example, **copy** *source* **running-config**).

- Modifications to the group of administrative users are logged; failure attempts for access to privileged EXEC mode ("enable" mode) are logged.

Note EAL Certification is not claimed by Cisco. These enhancements provide the groundwork for future certification.

The logging actions described above are disabled by default. To enable these logging characteristics, perform the task described in the "Configuring the Configuration Change Notification and Logging Feature" section in the "Configuration Change Notification and Logging" feature module.

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:**	Enters archive configuration mode.

	Command or Action	Purpose
	`Device(config)# archive`	
Step 4	**log config** **Example:** `Device(config-archive)# log config`	Enters configuration change logger configuration mode.
Step 5	**logging enable** **Example:** `Device(config-archive-log-config)# logging enable`	Enables the logging of configuration changes. • Logging of configuration changes is disabled by default.
Step 6	**logging size** *entries* **Example:** `Device(config-archive-log-config)# logging size 200`	(Optional) Specifies the maximum number of entries retained in the configuration log. • Valid values for the *entries* 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:** `Device(config-archive-log-config)# hidekeys`	(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:** `Device(config-archive-log-config)# notify syslog`	(Optional) Enables the sending of notifications of configuration changes to a remote syslog.
Step 9	**end** **Example:** `Device(config-archive-log-config)# end`	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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• 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** *entries* **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** *entries* **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.
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
```

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 *Managing Configuration Files Configuration Guide*
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 www.cisco.com/go/cfn. 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**.

CHAPTER 8

Configuration Partitioning

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

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

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

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

- Finding Feature Information, on page 79
- Information About Configuration Partitioning, on page 80
- How to Use the Configuration Partitioning Feature, on page 81
- Configuration Examples for Configuration Partitioning, on page 84
- Additional References, on page 93
- Feature Information for Configuration Partitioning, on page 95

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About Configuration Partitioning

System Running Configurations

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

Retrieving the Running Configuration for Display or Copy Operations

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

Benefits of Partitioning the Running Configuration

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

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

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

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

More granular access to configuration information offers important performance benefits for high-end routing platforms with very large configuration files, while also enhancing configuration management by allowing advanced configuration features to be implemented at a more granular level. Advanced configuration options include Cisco IOS software support for provisioning of customer services, Config Rollback, Config Locking, and configuration access control.

How to Use the Configuration Partitioning Feature

Displaying Configuration Partitions

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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

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

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

Example:

```
Router# show running-config partition ?
 config partition is TRUE
  access-list       All access-list configurations
  boot              All boot configurations
  class-map         All class-map configurations
  common            All remaining unregistered configurations
  global-cdp        All global cdp configurations
  interface         All Interface specific Configurations
  ip-as-path        All IP as-path configurations
  ip-community      All IP community list configurations
  ip-domain-list    All ip domain list configurations
  ip-prefix-list    All ip prefix-list configurations
  ip-static-routes  All IP static configurations
  line              All line mode configurations
  policy-map        All policy-map configurations
  route-map         All route-map configurations
  router            All routing configurations
  snmp              All SNMP configurations
  tacacs            All TACACS configurations
```

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

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

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

Example:

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

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

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

Example:

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

Disabling the Configuration Partitioning Feature

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**no parser config partition**
	Example:
	`Router(config)# no parser config partition`
	Example:
	`Disabling config partitioning`
	Example:
	`Router(config)#`
		Disables the configuration partitioning feature.

What to do next

What to Do Next

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

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

Configuration Examples for Configuration Partitioning

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

Displaying Configuration Partitions Example

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

Note The *part* argument can consist of multiple partition name keywords, as in **showrunning-configpartroutereigrp1**.

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

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

access-list 90 permit 0.0.0.0 1.2.3.5
```

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

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

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

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

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

global-cdp

cdp timer 20
cdp holdtime 100
gt3-7200-3#
gt3-7200-3# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively down down
Ethernet2/0                10.4.2.32       YES NVRAM  up                    up
Ethernet2/1                unassigned      YES NVRAM  administratively down down
Ethernet2/2                unassigned      YES NVRAM  administratively down down
Ethernet2/3                unassigned      YES NVRAM  administratively down down
Serial3/0                  unassigned      YES NVRAM  administratively down down
Serial3/1                  unassigned      YES NVRAM  administratively down down
Serial3/2                  unassigned      YES NVRAM  administratively down down
Serial3/3                  unassigned      YES NVRAM  administratively down down
Loopback0                  unassigned      YES NVRAM  administratively down down
Loopback234                unassigned      YES NVRAM  administratively down down
```

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

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

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

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

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

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

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

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

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

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

```
Building configuration...
Current configuration : 162 bytes
!
!
!
policy-map qwer
 description policy-map qwer.
  class xyz
    shape peak 8000 32 32
policy-map p1
policy-map sdf
  class abc
   set precedence 4
!
!
!
end
gt3-7200-3# show running-config partition route-map
Building configuration...
Current configuration : 65 bytes
!
!
!
route-map iop permit 10
!
route-map rty permit 10
!
!
end
gt3-7200-3#sh run part router bgp 1
Building configuration...
Current configuration : 111 bytes
!
!
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 distance bgp 2 2 2
 no auto-summary
!
!
end
gt3-7200-3#sh run part router egp ?
<0-65535>  Remote autonomous system number
gt3-7200-3#sh run part router egp 1
Building configuration...
Current configuration : 46 bytes
!
!
!
router egp 1
 timers egp 20 20
!
!
end
gt3-7200-3# show running-config partition router ?
  bgp       Border Gateway Protocol (BGP)
  egp       Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      ISO IS-IS
  iso-igrp  IGRP for OSI networks
  mobile    Mobile routes
  odr       On Demand stub Routes
  ospf      Open Shortest Path First (OSPF)
  rip       Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router eigrp ?
<1-65535>  Autonomous system number
gt3-7200-3# show running-config partition router eigrp 1
Building configuration...
Current configuration : 13 bytes
!
!
!
!
end
gt3-7200-3#
gt3-7200-3# sh run part router eigrp 2

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

Building configuration...
```

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

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

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

router mobile
router odr
router eigrp 2
router ospf 4
router iso-igrp
router isis qwe
router egp 1
router bgp 1
gt3-7200-3# show running-config partition router ?
  bgp       Border Gateway Protocol (BGP)
  egp       Exterior Gateway Protocol (EGP)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      ISO IS-IS
  iso-igrp  IGRP for OSI networks
  mobile    Mobile routes
  odr       On Demand stub Routes
  ospf      Open Shortest Path First (OSPF)
  rip       Routing Information Protocol (RIP)
gt3-7200-3# show running-config partition router ospf ?
  <1-65535>  Process ID
gt3-7200-3# show running-config partition router ospf 4
Building configuration...
Current configuration : 64 bytes
!
!
!
router ospf 4
 log-adjacency-changes
 distance 4
!
!
end
gt3-7200-3# sh run part service

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

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

Additional References

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

Related Documents

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

Related Topic	Document Title
Provisioning of customer services, Config Rollback, Config Locking, and configuration access control	Contextual Configuration Diff Utility
Configuration management--Config change logging.	Configuration Change Notification and Logging
Configuration management --Quick-save for config change logging[3].	Configuration Logger Persistency
Cisco IOS software configuration access control and config session locking ("Config Lock").	Exclusive Configuration Change Access and Access Session Locking

[3] The "Configuration Logger Persistency" feature allows saving just the commands entered since the last startup-config file was generated, rather than saving the entire startup configuration.

Standards

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

MIBs

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

RFCs

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

Technical Assistance

Description	Link
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 Partitioning

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for Configuration Partitioning

Feature Name	Releases	Feature Information
Configuration Partitioning	12.2(33)SRB 12.2(33)SB 12.2(33)SXI	The Configuration Partitioning feature provides modularization ("partitioning") of the running configuration state to provide granular access to the running configuration in Cisco IOS software. This feature is enabled by default in Cisco IOS software images that include this feature. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. The following sections provide information about this feature: • Information About Configuration Partitioning • How to Use the Configuration Partitioning Feature

CHAPTER 9

Configuration Versioning

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

- Finding Feature Information, on page 97
- Information About Configuration Versioning, on page 97
- How to Configure Configuration Versioning, on page 98
- Configuration Examples for Configuration Versioning, on page 102
- Additional References, on page 102
- Feature Information for Configuration Versioning, on page 103

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About Configuration Versioning

Configuration Archive

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

The **archive config** command allows you to save Cisco configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and optional time stamp) as each consecutive file is saved. This functionality provides consistent identification of saved configuration files. You can specify how many versions of the running configuration are kept in the archive. After the maximum number of files are saved in the archive, the oldest file is automatically deleted when the next, most recent file is saved. The **show archive** command displays information for all configuration files saved in the configuration archive.

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

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

How to Configure Configuration Versioning

Configuring the Characteristics of the Configuration Archive

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**archive** **Example:** `Device(config)# archive`	Enters archive configuration mode.
Step 4	**path** *url* **Example:** `Device(config-archive)# path bootflash:myconfig`	Specifies the location and filename prefix for the files in the configuration archive. • Depending on your hardware platform, the name of your file system might be different than shown in the example. **Note** If a directory is specified in the path instead of file, the directory name must be followed by a forward slash as follows: path flash:/*directory*/. The forward slash is not necessary after a filename; it is necessary only when specifying a directory.
Step 5	**maximum** *number* **Example:** `Device(config-archive)# maximum 14`	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the configuration archive. • The *number* argument is the maximum number of archive files of the running configuration to be saved in the configuration archive. The range is 1 to 14. The default is 10. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the configuration archive.
Step 6	**time-period** *minutes* **Example:** `Device(config-archive)# time-period 10`	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the configuration archive. • The *minutes* argument specifies how often, in minutes, to automatically save an archive file of the current running configuration in the configuration archive. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the configuration archive.
Step 7	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-archive)# end`	
Step 8	**archive config** **Example:** `Device# archive config`	Saves the current running configuration file to the configuration archive. **Note** You must configure the **path** command before using the **archive config** command.

Monitoring and Troubleshooting the Configuration

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **enable**

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

Example:

```
Device> enable
Device#
```

Step 2 **show archive**

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

Example:

```
Device# show archive

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

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

Example:

```
Device# show archive

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

Step 3 **debug archive versioning**

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

Example:

```
Device# debug archive versioning
Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file bootflash:myconfig-7
Jan  9 06:46:29.547: backup worked
```

Step 4 **debug archive config timestamp**

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

Example:

```
Device# debug archive config timestamp
Device# configure replace bootflash:myconfig force
Timing Debug Statistics for IOS Config Replace operation:
       Time to read file slot0:sample_2.cfg = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
Starting Pass 1
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:93
       Size of file        :2539
       Time taken for positive rollback pass = 320 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for negative incremental diffs pass = 59 msec (0 sec)
       Time taken by PI to apply changes = 0 msec (0 sec)
       Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
       Time taken for positive rollback pass = 0 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done
```

Step 5 **exit**

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

Example:

```
Device# exit
Device>
```

Configuration Examples for Configuration Versioning

Example: Creating a Configuration Archive

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

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Information about managing configuration files	“Managing Configuration Files” module in the *Managing Configuration Files Configuration Guide*
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 Versioning

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for Configuration Versioning

Feature Name	Releases	Feature Information
Configuration Versioning	12.2(25)S 12.2(33)SRA 12.3(7)T Cisco IOS XE Release 2.1	The Configuration Versioning feature allows you to maintain and manage backup copies of the Cisco running configuration on or off the device. The Configuration Replace feature uses the Configuration Versioning feature to provide a rollback to a saved copy of the running configuration. The following commands were introduced or modified: **archive config**, **debug archive versioning**, **log config**, **maximum**, **path** (archive configuration), **show archive**, **time-period**, **write-memory**.

CHAPTER 10

Configuration Rollback Confirmed Change

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

- Finding Feature Information, on page 105
- Information About Configuration Rollback Confirmed Change, on page 105
- How to Configure Configuration Rollback Confirmed Change, on page 106
- Configuration Examples for Configuration Rollback Confirmed Change, on page 108
- Additional References, on page 109
- Feature Information for Configuration Rollback Confirmed Change, on page 109

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Information About Configuration Rollback Confirmed Change

Configuration Rollback Confirmed Change Operation

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

The following steps outline how this process is achieved:

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

How to Configure Configuration Rollback Confirmed Change

Performing a Configuration Replace or Configuration Rollback Operation with Confirmation

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignorecase**] [**revert trigger** [**error**] [**timer** *minutes*] | **time** *minutes*] **Example:** `Device# configure replace bootflash:myconfig-1 list time 30`	Replaces the current running configuration file with a saved configuration file. • *target-url*—Specifies a URL (accessible by the Cisco file system) of the saved configuration file that is to replace the current running configuration, such as the configuration file created by using the **archive config** command. Depending on your hardware platform, the name of your file system might be different than shown in the example.

	Command or Action	Purpose

		• **nolock**—Disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation.
		• **list**—Displays a list of the command lines applied by the Cisco software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed.
		• **force**—Replaces the current running configuration file with the specified saved configuration file without prompting you for confirmation.
		• **ignorecase**—Allows the configuration to ignore the case of the confirmation command.
		• **time** *minutes*—Specifies the time (in minutes) within which you must enter the **configure confirm** command to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the **configure replace** command).
		• **revert trigger**—Sets the following triggers for reverting to the original configuration:
		  • **error**—Reverts to the original configuration upon error.
		  • **timer** *minutes*—Reverts to the original configuration if the specified time elapses.

Step 3	**configure revert** {**now** | **timer** {*minutes* | **idle** *minutes*}}
	Example:
	`Device# configure revert now`
		(Optional) Cancels the timed rollback and triggers the rollback immediately or resets parameters for the timed rollback.
		• **now**—Triggers the rollback immediately.
		• **timer**—Resets the configuration revert timer.
		  • Use the *minutes* 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.

	Command or Action	Purpose
Step 4	**configure confirm** **Example:** `Device# configure confirm`	(Optional) Confirms replacement of the current running configuration file with a saved configuration file. **Note** Use this command only if the **time** *minutes* keyword and argument of the **configure replace** command are specified.
Step 5	**exit** **Example:** `Device# exit`	Exits to user EXEC mode.

Configuration Examples for Configuration Rollback Confirmed Change

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

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

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

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

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Information about managing configuration files	"Managing Configuration Files" module in the *Managing Configuration Files Configuration Guide*
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 Rollback Confirmed Change

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for Configuration Rollback Confirmed Change

Feature Name	Releases	Feature Information
Configuration Rollback Confirmed Change	12.2(33)SB 12.2(33)SRC 12.2(33)SXI 12.4(20)T Cisco IOS XE Release 2.1	The Configuration Rollback Confirmed Change feature allows configuration changes to be performed with an optional requirement that they be confirmed. 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 commands were introduced or modified: **configure confirm**, **configure replace**, **configure revert**, **configure terminal**.

CHAPTER 11

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, on page 111
- Prerequisites for Configuration Logger Persistency, on page 111
- Information About Configuration Logger Persistency, on page 112
- How to Configure the Configuration Logger Persistency Feature, on page 113
- Configuration Examples for the Configuration Logger Persistency Feature, on page 116
- Additional References, on page 116
- Feature Information for Configuration Logger Persistency, on page 117
- Glossary, on page 118

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 **copy***url* **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 persistent** *size threshold*
8. **logging size** *entries*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.

	Command or Action	Purpose
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	**log config** **Example:** `Router(config-archive)# log config`	Enters archive configuration-log configuration mode.
Step 5	**logging persistent auto manual** **Example:** `Router(config-archive-log-cfg)# logging persistent auto`	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:** `Router(config-archive-log-cfg)# logging persistent reload`	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 persistent** *size threshold* **Example:** `Router(config-archive-log-cfg)# logging persistent size threshold`	Specifies the disk space size for writing log messages in the configuration logger database; triggers an alert on the console or syslog server when the log size exceeds the threshold (specified in percentage).
Step 8	**logging size** *entries* **Example:** `Router(config-archive-log-cfg)# logging size 10`	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
```

ConfigurationExamplesfortheConfigurationLoggerPersistency 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 persistent size 16384 threshold 10
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	*Cisco IOS Configuration Fundamentals Command Reference*

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: 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.

CHAPTER 12

Software Maintenance Upgrade

The Software Maintenance Upgrade (SMU) is a package that can be installed on a system to provide a patch fix or security resolution to a released image.

- Information About Software Maintenance Upgrade, on page 119
- Restrictions for Software Maintenance Upgrade, on page 120
- How to Configure Software Maintenance Upgrade, on page 121
- Configuration Examples for Software Maintenance Upgrade, on page 122
- Additional References for Software Maintenance Upgrade, on page 128
- Feature Information for Software Maintenance Upgrade, on page 128

Information About Software Maintenance Upgrade

Software Maintenance Upgrade

The Software Maintenance Upgrade (SMU) is a package that can be installed on a system to provide a patch fix or security resolution to a released image.

An SMU package is provided on a per release and per component basis and is specific to the platform.

An SMU provides a significant benefit over classic IOS software as it allows you to address the network issue quickly while reducing the time and scope of the testing required. The Cisco IOS XE platform internally validates the SMU compatibility and does not allow you to install non-compatible SMUs.

All SMUs are integrated into the subsequent Cisco IOS XE software maintenance releases. An SMU is an independent and self-sufficient package and it does not have any prerequisites or dependencies. You can choose which SMUs to install or uninstall in any order.

Starting from Cisco IOS XE Everest 16.6.1, SMUs are supported only on Extended Maintenance releases and for the full lifecycle of the underlying software release.

The following are three basic steps to install an SMU:

- Addition of the SMU to the filesystem
- Activation of the SMU onto the system
- Committing the SMU change so it is persistent across reloads

Supported Platforms

Starting Cisco IOS XE Everest 16.6.1, the following platforms are supported for Software Maintenance Upgrade:

- Cisco ASR 1000 Series Aggregate Services Routers (ASR1001-X, ASR1002-X, ASR1001-HX, ASR1002-HX, ASR1000-RP2, ASR1000-RP3)
- Cisco ISR 4000 Series Integrated Services Routers (ISR4351, ISR4331, ISR4431, ISR4321, ISR4451)
- Cisco CSR 1000v Series Cloud Services Routers
- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches

Software Maintenance Upgrade Package

An SMU package contains metadata and the fix for the reported issue that the SMU is requested for.

Software Maintenance Upgrade Workflow

The SMU process is initiated with a request to the SMU committee. Contact your customer support to raise an SMU request.

At release time, SMU packages are posted to the Cisco Software Download page along with the following information:

- Defect addressed
- Type of defect -PSIRT and so on

SMU Reload

The SMU type describes the effect to the system after installing the SMU. SMUs can be non-traffic affecting or can result in device reload or switchover.

A cold reload of the system requires the complete reload of the operating system. This action effects traffic flow for the duration of the reload (~5 min currently). Reload ensures that all processes are started with the correct libraries and files that are installed as part of the SMU.

Restrictions for Software Maintenance Upgrade

- Switchover for In Service Software Upgrade (ISSU) is not supported.

How to Configure Software Maintenance Upgrade

Adding, Activating, Committing an SMU

To install an SMU package, copy the downloaded SMU package to the relevant directory on your device. Use the following commands to add, activate, and commit an SMU:

- **install add**: Runs base compatibility checks on a file to ensure that the SMU package is supported on the platform. It also adds an entry in the package/SMU .sta file, so that its status can be monitored/maintained from here on. The install add command takes the following input: package file location and downloading method (tftp, ftp, and so on)
- **install active**: Runs compatibility checks, installs the package, updates package status details. For restartable package it either triggers the appropriate post-install scripts to restart necessary processes or for non-restartable packages it will trigger a reload.
- **install commit**: Commits the activation changes to be persistent across all reloads. The commit can be done after activation while the system is up, or after the first reload. If a package was activated but not committed, it will remain active after the first reload, but not after the second reload.

Perform the following configuration to add, activate, and commit an SMU:

```
enable
install add file bootflash:isr4300-universalk9.BLD_
SMU_LATEST_20170128_040557.1.CSCxxx.SSA.smu.bin

show install summary  // Shows the installed SMU package as inactive package in the command
 output

install activate file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxx.SSA.smu.bin

show version // Shows the image version tagged with the  “SMU Patched” phrase

show install summary // Shows the installed SMU package as an active package in the command
 output

install commit

show install summary // Shows the installed SMU package as a committed package in the command
 output.
```

Rolling Back, Deactivating, or Removing an SMU

Use the following commands to roll back, deactivate, and remove an SMU:

- **`install rollback`**: Returns the device to the previous installation state. This rollback requires a reload.
- **`install deactivate`**: Deactivates an active package, updates the package status, and triggers a process restart or a reload.
- **`install remove`**: Removes all or specified inactive SMU packages from the file system.

Perform the following tasks to roll back, deactivate, or remove an SMU:

```
enable
install rollback to committed

install deactivate file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxXXXX.SSA.smu.bin


install remove file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxx.SSA.smu.bin
```

Configuration Examples for Software Maintenance Upgrade

Example: Adding, Activating, and Committing an SMU

Adding, Activating, and Committing an SMU

The following example shows the workflow for adding, activating, and committing an SMU

```
Device# install add file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxXXXX.SSA.smu.bin
install_add: START Tue Aug  1 04:22:48 UTC 2017
install_add: Adding SMU


*Aug  1 04:22:54.492: %IOSXE-5-PLATFORM:  SIP2: Aug  1 04:22:54 install_engine.sh:
%INSTALL-5-INSTALL_START_INFO: Started install add
bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin--- Starting SMU Add operation
 ---
Performing SMU_ADD on Active/Standby
  [R0] SMU_ADD package(s) on R0
  [R0] Finished SMU_ADD on R0
Checking status of SMU_ADD on [R0]
SMU_ADD: Passed on [R0]
Finished SMU Add operation

SUCCESS: install_add  Tue Aug  1 04:23:10 UTC 2017


Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device# install activate file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxx.SSA.smu.bin

install_activate: START Tue Aug  1 04:24:42 UTC 2017
install_activate: Activating SMU

*Aug  1 04:24:48.682: %IOSXE-5-PLATFORM:  SIP2: Aug  1 04:24:48 install_engine.sh:
%INSTALL-5-INSTALL_START_INFO: Started install activate
```

```
bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
This operation requires a reload of the system. Do you want to proceed? [y/n]y

--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on Active/Standby
  [R0] SMU_ACTIVATE package(s) on R0
    DMP package.
  [R0] Finished SMU_ACTIVATE on R0
Checking status of SMU_ACTIVATE on [R0]
SMU_ACTIVATE: Passed on [R0]
Finished SMU Activate operation

install_activate: Reloading the box to complete activation of the SMU...
install_activate will reload the system now!
Aug  1 04:25:36
Aug  1 04:25:45.742 RP0/0: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate
SMU bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin

<after reload>

Device# show version
Cisco IOS XE Software, Version 16.06.01 - SMU-PATCHED
Cisco IOS Software [Everest], ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.6.1,
 RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Sat 22-Jul-17 05:55 by mcpre

...

Active SMU Information:
  State (St):  C - Committed, U - Uncommitted
--------------------------------------------------------------------------------
Type  Defect_ID    Version   St Filename
--------------------------------------------------------------------------------
SMU   CSCxxXXXXX     16.6.1.0. U  isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu
--------------------------------------------------------------------------------
cisco ISR4351/K9 (2RU) processor with 7941107K/6147K bytes of memory.
Processor board ID FLM2007W0MJ
3 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
14659583K bytes of flash memory at bootflash:.
0K bytes of WebUI ODM Files at webui:.

Configuration register is 0x0

Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   U    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device# show install active
 R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
```

```
SMU   U    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device# install commit
install_commit: START Tue Aug  1 04:48:03 UTC 2017
install_commit: Committing SMU


*Aug  1 04:48:10.042: %IOSXE-5-PLATFORM:  SIP2: Aug  1 04:48:10 install_engine.sh:
%INSTALL-5-INSTALL_START_INFO: Started install commit--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [R0] SMU_COMMIT package(s) on R0
  [R0] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation
SUCCESS: install_commit  Tue Aug  1 04:48:33 UTC 2017

Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0
```

Example: Rolling Back, Deactivating, or Removing an SMU

Example: Rolling back, Deactivating, or Removing an SMU

```
Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device#show install rollback
ID      Label          Description
-----------------------------------------------
4       No Label        No Description

Device# install rollback to committed

install_rollback: START Tue Aug  1 05:00:37 UTC 2017

*Aug  1 05:00:44.038: %IOSXE-5-PLATFORM:  SIP2: Aug  1 05:00:44 install_engine.sh:
%INSTALL-5-INSTALL_START_INFO: Started install rollbackinstall_rollback: Rolling back SMU

This operation requires a reload of the system. Do you want to proceed? [y/n]y

--- Starting SMU Rollback operation ---
Performing SMU_ROLLBACK on Active/Standby
  [R0] SMU_ROLLBACK package(s) on R0
  [R0] Finished SMU_ROLLBACK on R0
Checking status of SMU_ROLLBACK on [R0]
SMU_ROLLBACK: Passed on [R0]
Finished SMU Rollback operation

install_rollback will reload the system now!
Aug  1 05:01:40.43
Aug  1 05:01:53.558 RP0/0: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install rollback
SMU

<after reload>


Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    16.6.1.0


//install deactivate: Deactivates an active package and triggers a process restart or a
reload.

Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device# install deactivate file
bootflash:isr4300-universalk9.BLD_SMU_LATEST_20170128_040557.1.CSCxxXXXX.SSA.smu.bin
install_deactivate: START Tue Aug  1 05:28:47 UTC 2017
install_deactivate: Deactivating SMU

This operation requires a reload of the system. Do you want to proceed? [y/n]y

--- Starting SMU Deactivate operation ---
Performing SMU_DEACTIVATE on Active/Standby
  [R0] SMU_DEACTIVATE package(s) on R0
    DMP package.
  [R0] Finished SMU_DEACTIVATE on R0
Checking status of SMU_DEACTIVATE on [R0]
SMU_DEACTIVATE: Passed on [R0]
Finished SMU Deactivate operation

install_deactivate: Reloading the box to complete activation of the SMU...
install_deactivate will reload the system now!

<after reload>


Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
```

```
--------------------------------------------------------------------------------
SMU   D    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device#install commit
install_commit: START Tue Aug  1 05:39:29 UTC 2017
install_commit: Committing SMU


*Aug  1 05:39:35.222: %IOSXE-5-PLATFORM:  SIP2: Aug  1 05:39:35 install_engine.sh:
%INSTALL-5-INSTALL_START_INFO: Started install commit--- Starting SMU Commit operation ---
Performing SMU_COMMIT on Active/Standby
  [R0] SMU_COMMIT package(s) on R0
  [R0] Finished SMU_COMMIT on R0
Checking status of SMU_COMMIT on [R0]
SMU_COMMIT: Passed on [R0]
Finished SMU Commit operation

SUCCESS: install_commit  Tue Aug  1 05:39:58 UTC 2017
Completed install commit SMU

Device#show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0


//install remove: Deletes the inactive SMU file from the file system.

Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0

Device#install remove file bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
install_remove: START Tue Aug  1 05:43:22 UTC 2017
install_remove: Removing SMU

--- Starting SMU Remove operation ---
Performing SMU_REMOVE on Active/Standby
  [R0] SMU_REMOVE package(s) on R0
  [R0] Finished SMU_REMOVE on R0
Checking status of SMU_REMOVE on [R0]
SMU_REMOVE: Passed on [R0]
Finished SMU Remove operation

SUCCESS: install_remove  Tue Aug  1 05:43:43 UTC 2017

//Remove inactive: Deletes all inactive packages from the file system


Device#show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I    bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
IMG   C    16.6.1.0


Device#install remove inactive
install_remove: START Tue Aug  1 05:52:31 UTC 2017
Cleaning up unnecessary package files
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
    isr4300-universalk9.16.06.01.SPA.bin
      File is in use, will not delete.
    packages.conf
      File is in use, will not delete.
  done.

The following files will be deleted:
[R0]:
/bootflash/isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin

Do you want to remove the above files? [y/n]y
[R0]:
Deleting file bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin ... done.
SUCCESS: Files deleted.
--- Starting Post_Remove_Cleanup ---
Performing Post_Remove_Cleanup on Active/Standby
  [R0] Post_Remove_Cleanup package(s) on R0
  [R0] Finished Post_Remove_Cleanup on R0
Checking status of Post_Remove_Cleanup on [R0]
Post_Remove_Cleanup: Passed on [R0]
Finished Post_Remove_Cleanup

SUCCESS: install_remove  Tue Aug  1 05:53:19 UTC 2017


///Show install package

Device#show install package bootflash:isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
Name: isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin
Version: 16.6.1.0.202.1500742946..Everest
Platform: ISR4300
Package Type: SMU
Defect ID: CSCxxXXXXX
Package State: Not Installed
Supersedes List: {}
SMU ID: 0
SMU Type: reload
SMU Compatible with Version: 16.6.1.0.202


//Show install log
Device#show install log
[0|install_op_boot]: START Tue Aug  1 05:34:59 Universal 2017
[0|install_op_boot(INFO, )]: SMU
/bootflash/isr4300-universalk9.16.06.01.CSCxxXXXXX.SPA.smu.bin will be activated upon reload.
[0|install_op_boot]: END SUCCESS  Tue Aug  1 05:35:06 Universal 2017
```

Additional References for Software Maintenance Upgrade

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Feature Information for Software Maintenance Upgrade

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12: Feature Information for Software Maintenance Upgrade

Feature Name	Releases	Feature Information
Software Maintenance Upgrade	Cisco IOS XE Everest 16.6.1.	The Software Maintenance Upgrade (SMU) is a package that can be installed on a system in order to provide a patch fix or security resolution to a released image The following commands were introduced or modified: **install**, **show install**