CHAPTER 4

Managing the ACE Software

This chapter describes how to manage the software running on the Cisco Application Control Engine (ACE) module and contains the following major sections:

Saving Configuration Files

Upon startup, the ACE loads the startup-configuration file stored in Flash memory (nonvolatile memory) to the running-configuration file stored in RAM (volatile memory). When you partition your ACE into multiple contexts, each context contains its own startup-configuration file.

Flash memory stores the startup-configuration files for each existing context. When you create a new context, the ACE creates a new context directory in Flash memory to store the context-specific startup-configuration files. When you copy a configuration file from the ACE, you create a copy of the configuration information of the context from where you executed the command.

When you make configuration changes, the ACE places those changes in a virtual running-configuration file called the running-config, which is associated with the context that you are working in. When you enter a CLI command, the change is made only to the running-configuration file in volatile memory. Before you log out or reboot the ACE, copy the contents of the running-configuration file to the startup-configuration file (startup-config) to save configuration changes for the current context to Flash memory. The ACE uses the startup-configuration file on subsequent reboots.

This section contains the following topics:

Saving the Configuration File in Flash Memory

This section describes how to save the contents of the running-configuration file in RAM (volatile memory) to the startup-configuration file for the current context in Flash memory (nonvolatile memory) on the ACE.

Detailed Steps

Command	Purpose
copy running-config startup-config **Example:** host1/Admin# copy running-config startup-config	Copies the contents of the running-configuration file to the startup-configuration file.
write memory [**all**] **Example:** host1/Admin# write memory all	Copies the contents of the running-configuration file to the startup-configuration file. The optional **all** keyword saves configurations for all existing contexts. This keyword is available only in the Admin context. When used without the **all** keyword, this command copies the contents of the running-configuration file for the current context to the startup-configuration file. **Note** After you save the contents of the running-configuration file for the current user context to the startup-configuration file, you should also save the changes to the Admin context startup-configuration file, which contains all configurations that are used to create each user context.

Saving Configuration Files to a Remote Server

This section describes how to save the running-configuration file or startup-configuration file to a remote server using File Transfer Protocol (FTP), Secure File Transfer Protocol (SFTP), or Trivial Transfer Protocol (TFTP). The copy serves as a backup file for the running-configuration file or startup-configuration file for the current context. Before installing or migrating to a new software version, back up the ACE startup-configuration file to a remote server using FTP, SFTP, or TFTP. When you name the backup file, we recommend that you name it in such a way that you can easily tell the context source of the file (for example, running-config-ctx1, startup-config-ctx1).

Detailed Steps

Command	Purpose
copy {**running-config** | **startup-config**} {**ftp:***//server/path*[*/filename*] | **sftp:**//[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} **Example:** host1/Admin# copy running-config ftp://192.168.1.2/running-config_Adminctx Enter username[]? user1 Enter the file transfer mode[bin/ascii]: [bin] Password: password1 Passive mode on. Hash mark printing on (1024 bytes/hash mark). ####	Saves the running-configuration file or startup-configuration file to a remote server using FTP, SFTP, or FTP. The keywords, arguments, and options are as follows: • **running-config**—Specifies the running-configuration file currently residing on the ACE in volatile memory. • **startup-config**—Specifies the startup-configuration file currently residing on the ACE in Flash memory. • **ftp://***server*/*path*[*/filename*]—Specifies the FTP network server and, optionally, the renamed configuration file. When using FTP, the bin (binary) file transfer mode is intended for transferring compiled files (executables). The ascii file transfer mode is intended for transferring text files, such as config files. The default selection of bin should be sufficient in all cases when copying files to a remote FTP server. • **sftp://**[*username@*]*server*/*path*[*/filename*]—Specifies the SFTP network server and, optionally, the renamed configuration file. • **tftp://***server*[**:***port*]**/***path*[*/filename*]—Specifies the TFTP network server and, optionally, the renamed configuration file. When you select a destination file system using **ftp:**, **sftp:**, or **tftp:**, the ACE performs the following tasks: • Prompts you for your username and password if the destination file system requires user authentication. • Prompts you for the server information if you do not provide the information with the command. • Copies the file to the root directory of the destination file system if you do not provide the path information.

Copying the Configuration File to the disk0: File System

This section describes how to copy the running-configuration file or the startup-configuration file to the disk0: file system in Flash memory on the ACE.

Detailed Steps

Command	Purpose
copy {**running-config** | **startup-config**} **disk0:**[*path/*]*filename* **Example:** host1/Admin# copy running-config disk0:running-config_copy	Copies either the running configuration of the startup configuration to a file on the disk0: file system in Flash memory. The keywords and arguments are as follows: • **running-config**—Specifies the running-configuration file currently residing on the ACE in RAM (volatile memory). • **startup-config**—Specifies the startup-configuration file currently residing on the ACE in Flash memory (nonvolatile memory). • [*path/*]*filename*—Path in the disk0: file system. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system.

Merging the Startup-Configuration File with the Running-Configuration File

This section describes how to merge the contents of the startup-configuration file into the running-configuration file. This process copies any additional configurations from the startup-configuration file into the running-configuration file. If any common commands exist in both files, the startup-configuration file overwrites the attributes in the running-configuration file.

Detailed Steps

Command	Purpose
copy startup-config running-config **Example:** host1/Admin# copy startup-config running-config	Merges the contents of the startup-configuration file into the running-configuration file.

Displaying Configuration File Content

To display the content of the running- and startup-configuration files, perform one of the following tasks:

Command	Purpose
show running-config [**aaa** | **access-list** | **action-list** | **class-map** | **context** | **dhcp** | **domain** | **ft** | **interface** | o**bject-group** | **parameter-map** | **policy-map** | **probe** | **resource-class** | **role** | **rserver** | **serverfarm** | **sticky**]	Displays the contents of the running configuration associated with the current context. Configuration entries within each mode appear in the chronological order in which you configure the ACE. The ACE does not display default configurations in the ACE running-configuration file. The keywords and options are as follows: • **aaa**—(Optional) Displays AAA information. • **access-list**—(Optional) Displays access control list (ACL) information. • **action-list**—(Optional) Displays action-list information. • **class-map**—(Optional) Displays all class maps configured for the current context. The ACE also displays configuration information for each class map. • **context**—(Optional) Displays the contexts configured on the ACE. The ACE also displays the resource class (member) assigned to each context. The **context** keyword works only from within the Admin context. • **dhcp**—(Optional) Displays Dynamic Host Configuration Protocol (DHCP) information. • **domain**—(Optional) Displays the domains configured for the current context. The ACE also displays configuration information for each domain listed. • **ft**—(Optional) Displays the redundancy or fault-tolerance (FT) configurations configured for the current context. The ACE also displays configuration information for each FT configuration. • **interface**—(Optional) Displays interface information. • **object-group**—(Optional) Displays ACL object-group information. • **parameter-map**—(Optional) Displays parameter map information. • **policy-map**—(Optional) Displays policy map information. • **probe**—(Optional) Displays probe information. • **resource-class**—(Optional) Displays resource class information. • **role**—(Optional) Displays the roles configured for the current context. The ACE also displays configuration information for each role. • **rserver**—(Optional) Displays real server information. • **serverfarm**—(Optional) Displays serverfarm information. • **sticky**—(Optional) Displays sticky information.
write terminal	Displays the contents of the running configuration associated with the current context. The **write terminal** command is equivalent to the **show running-config** command.

Command	Purpose
invoke context *context_name* **show running-config**	Displays the running-configuration file of a user context from the Admin context. The *context_name* argument is the name of the user context.
show startup-config	Displays the contents of the startup configuration associated with the current context.

Clearing the Startup-Configuration File

This section describes how to clear the contents of the ACE startup-configuration file of the current context in Flash memory. Both commands reset the startup-configuration file to the default settings and take effect immediately.

Restrictions

The **clear startup-config** and **write erase** commands used to clear the contents of the ACE startup-configuration file of the current context in Flash memory include the following restrictions:

- These commands do not affect the following items:
  - Running-configuration file
  - Boot variables, such as config-register and boot system settings

The commands do not remove the following items from the ACE startup-configuration file:

- License files—To remove license files, use the **license uninstall** *filename* command (see the "Removing a License Bundle or All License Bundles from the ACE" section on page 3-7.).
- Crypto files—To remove crypto files, use the **crypto delete** *filename* or the **crypto delete all** command (see the *Cisco Application Control Engine Module SSL Configuration Guide*).

Detailed Steps

	Command	Purpose
Step 1	**copy startup-config** {**ftp:**//*server/path*[/*filename*] | **sftp:**//[*username*@]*server/path*[/*filename*] | **tftp:**//*server*[:*port*]/*path*[/*filename*]} **Example:** host1/Admin# copy startup-config ftp://192.168.1.2/startup-config_Adminctx	(Optional) Creates a backup of your current startup-configuration file on a remote server. For details about using this command, see the "Saving Configuration Files to a Remote Server" section.
Step 2	**clear startup-config** **Example:** host1/Admin# clear startup-config	Clears the contents of the startup-configuration file and resets it to the default settings.
	write erase **Example:** host1/Admin# write erase	

	Command	Purpose
Step 3	**copy running-config startup-config** **Example:** host1/Admin# copy running-config startup-config	(Optional) Recovers a copy of an startup configuration by copying the contents of the existing running-configuration file to the startup-configuration file.
	copy {**ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} **startup-config** **Example:** host1/Admin# copy ftp://192.168.1.2/startup-config_Adminctx startup-config	(Optional) Recovers a copy of an existing startup configuration saved on a remote server. For details about using this command, see the "Copying Configuration Files from a Remote Server" section.

Copying Configuration Files from a Remote Server

This section describes how to configure the ACE by downloading a copy of a running-configuration file or startup-configuration file from a remote server. When you copy the backup configuration file to the ACE, you copy the configuration information to the context from where you initially executed the **copy** command.

Prerequisites

This topics includes the following prerequisites:

- You know the location of the configuration file to be loaded from the remote server.
- The configuration file permissions are set to world-read.
- The ACE has a route to the remote server. The ACE and the remote server must be in the same subnetwork if you do not have a router or default gateway to route the traffic between subnets. To check connectivity to the remote server, use the **ping** or **traceroute** command in Exec mode. See the *Cisco Application Control Engine Module Routing and Bridging Configuration Guide* for details on how to use the **ping** and **traceroute** commands.
- Ensure that the configuration file is appropriate for use in the current context. For example, you would copy the backup configuration file startup-config-ctx1 to context 1.

Detailed Steps

Command	Purpose
copy {**ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} {**running-config** | **startup-config**} **Example:** host1/Admin# copy ftp://192.168.1.2/startup-config_Adminctx startup-config	Configures the ACE using a running-configuration file or startup-configuration file downloaded from a remote server. For details about using this command, see the "Copying Configuration Files from a Remote Server" section.

Displaying the Configuration Download Progress Status

This section describes how to display the progress of a configuration download when a large configuration file in the ACE has been applied to a context.

When you apply changes to a configuration file, the ACE downloads the configuration to its data plane. When you perform incremental changes, such as copying and pasting commands in a configuration, the ACE immediately performs the configuration download and does not display any terminal messages at the start or end of the download.

However, in the following situations, the ACE defers the configuration download until the entire configuration is applied to the context:

- A startup configuration at boot time
- Copying of the configuration to the running-configuration file
- A checkpoint rollback

At the start of the deferred download, the ACE displays the following message on all terminals that are logged into the context including a terminal that you log into for the context before the download is done:

```
Processing has started for applied config
```

During the download, the ACE locks the context and denies any configuration changes until the download is completed.

Note We recommend that you do not execute any configuration commands during the deferred download. The ACE does not deny you from entering configuration changes. But the changes will not occur until the download is completed. If the command times out during the download, the following message appears:

```
Config application in progress. This command is queued to the system.
```

The ACE does not queue the command immediately, however, the ACE processes and executes the command when the download is completed even if the command times out.

You can execute the **show download information** command to monitor the progress of the download. You can also execute **show** commands that do not have interaction with the configuration manager (cfgmgr). For example, these commands include the **show acl-merge**, **show interface**, **show context**, **show crypto files,** and **show fifo** commands.

The **show** commands that have interaction with the cfgmgr do not work when the download occurs. For example, these commands include the **show access-list**, **show conn**, **show domain**, **show running-config**, and **show service-policy** commands. If you execute a cfgmgr show command during the download, the following error message occurs:

```
System Busy: Config application in progress
```

At the end of the deferred download, the ACE displays the follow message on all terminals that are logged into the context:

```
Processing has finished for applied config
```

To display the progress status of the configuration download on a context, perform the following task:

Command	Purpose
show download information [**all**] [**summary**]} **Example:** `host1/Admin# show download information all`	Displays the state of the configuration download for each interface on the context. If no option is included with this command, the status information for all interfaces in the current context is displayed. The options are as follows: • **all** —Displays the configuration download status for all interfaces on all contexts. This option is available in the Admin context. • **summary**—Displays the summary status of the download information for the context. When you include the **all** option with the **summary** option, the download summary status for all contexts is displayed. See Table 4-1 for information on the download states that the Download-status field displays.

Table 4-1 describes the fields that appear in the **show download information** command output.

Table 4-1 Field Descriptions for the show download information command

Field	Description
Context	Name of the context.
Interface	Number of the interface on the context. This field is not displayed with the **summary** option.
Download-Status	State of the configuration download. With no option or the **all** option, the possible states are as follows: • Pending—The interface has been updated but the update has not been downloaded. • In Progress—The interface download is in progress. • Completed—The interface download is completed. • Pending/Deleted—The interface has been deleted but it has not been downloaded. • In progress/Deleted—The interface has been deleted and the download is in progress. With the **summary** option, the possible states are as follows: • Completed—All of the interfaces have a status of Completed. • Pending—One or more of the interfaces are in the Pending state and the rest of the interfaces are in the Completed state. • In Progress—One or more interfaces are in the Progress state and the rest of the interfaces are in the Completed or Pending state.

Using the File System on the ACE

This section describes how use the ACE file system. Flash memory stores the operating system, startup-configuration files, software licenses, core dump files, system message log files, SSL certificates and keys, probe scripts, and other data on the ACE. Flash memory comprises a number of individual file systems, or partitions, that include this data.

The ACE contains the following file systems, or partitions:

- **disk0:**—Contains all startup-configuration files, software licenses, system message log files, SSL certificates and keys, and user-generated data for all existing contexts on the ACE.
- **image:**—Contains the system software images.
- **core:**—Contains the core files generated after each time that the ACE becomes unresponsive.
- **probe:**—Contains the Cisco-supplied scripts. For more information about these scripts, see the *Cisco Application Control Engine Module Server Load-Balancing Configuration Guide*. Both the Admin context and user contexts support the probe: directory.
- **volatile:**—Contains the files residing in the temporary (volatile:) directory. The volatile: directory provides temporary storage; files in temporary storage are erased when the ACE reboots.

The Admin context supports all five file systems in the ACE. The user context supports only the disk0:, probe:, and volatile: file systems.

When you create a new context, the ACE creates a new context directory in Flash memory to store context-specific data such as startup-configuration files.

The ACE provides a number of useful commands to help you manage software configuration and image and files.This section contains the following topics that will help you to manage files on the ACE:

Copying Files

This section describes how create copies of a file on the ACE and how to copy files to and from the ACE. This section contains the following topics:

Copying Files Between Directories in the disk0: File System on the ACE

This section describes how to copy a file from one directory in the disk0: file system of Flash memory to another directory in disk0:.

Detailed Steps

	Command	Purpose
Step 1	**dir disk0:** **Example:** host1/Admin# dir disk0:	(Optional) Displays the contents of the disk0: file system.
Step 2	**copy disk0:**[*path/*]*filename1* {**disk0:**[*path*]*filename2*} **Example:** host1/Admin# copy disk0:samplefile disk0:MYSTORAGE/SAMPLEFILE	Copies a file from one directory in the disk0: file system of Flash memory to another directory in disk0:. The keywords and arguments are as follows: • [*path/*]*filename1*—Name of the file to copy. Use the **dir disk0:** command to view the files available in the disk0: file system. If you do not provide the optional path, the ACE copies the file from the root directory on the disk0: file system. • **disk0:**[*path*]*filename2*—Specifies the file destination in the disk0: directory of the current context. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system.

Copying Licenses

This section describes how to create a backup license for the ACE licenses in .tar format and copy it to the disk0: file system. To protect your license files, we recommend that you back up your license files to the ACE Flash memory as tar files.

Detailed Steps

Command	Purpose
copy licenses disk0:[*path/*]*filename*.**tar** **Example:** host1/Admin# copy licenses disk0:mylicenses.tar	Creates a backup license for the ACE licenses in .tar format and copies it to the disk0: file system. The keyword and argument are as follows: • **disk0:**—Specifies that the backup license file is copied to the disk0: file system. • [*path/*]*filename*.**tar**—Destination filename for the backup licenses. The destination filename must have a .tar file extension. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system.
untar disk0:[*path/*]*filename*.**tar** **Example:** host1/Admin# copy licenses disk0:mylicenses.tar	(Optional) Untars the backup file and reinstalls it if you accidently remove or lose the license on the ACE (see the "Untarring Files in the disk0: File System" section).

Copying a Packet Capture Buffer

This section describes how to copy an existing packet capture buffer to the disk0: file system.

Detailed Steps

Command	Purpose
copy capture *capture_name* **disk0:**[*path/*]*destination_name* **Example:** host1/Admin# copy capture packet_capture_Jan_17_07 disk0:capture_Jan_17_07	Copies an existing packet capture buffer to the disk0: file system. The keywords, arguments, and options are as follows: • *capture_name*—Name of the packet capture buffer on Flash memory. Specify a text string from 1 to 64 alphanumeric characters. If necessary, use the **show capture** command to view the files available in the disk0: file system. This list includes the name of existing packet capture buffers. • **disk0:**—Specifies that the buffer is copied to the disk0: file system. • [*path/*]*destination_name*—Destination path (optional) and name for the packet capture buffer. Specify a text string from 1 to 80 alphanumeric characters. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system.

Copying a Scripted Probe File

This section describes how to copy a scripted probe file from the probe: directory to another directory on the disk0:file system on the ACE or a remote server using FTP, SFTP, or TFTP.

Restrictions

You cannot copy a scripted probe file to the probe: directory on the ACE.

The **copy probe:** command is available only in the Admin context.

Detailed Steps

Command	Purpose
copy probe:*filename* {**disk0:**[[*path*/]*filename*] | **ftp:**//*server*/*path*[/*filename*] | **sftp:**//[*username*@]*server*/*path*[/*filename*] | **tftp:**//*server*[:*port*]/*path*[/*filename*]} **Example:** host1/Admin# copy probe: disk0: Enter source filename[]? LDAP_PROBE_SCRIPT Destination filename[]?:[LDAP_PROBE_SCRIPT] host1/Admin#	Copies a file from the probe: directory to the disk0: file system on the ACE or a remote server using FTP, SFTP, or TFTP. The keywords, arguments, and options are as follows: • **probe:***filename*—Specifies the scripted probe file residing on the ACE. Use the **dir probe:** command to view the files available in the probe: directory. • **disk0:**[*path*/]*filename***—**Specifies a location and filename in the disk0: file system. • **ftp://***server***/***path*[**/***filename*]—Specifies the FTP network server and, optionally, the renamed file. When using FTP, the bin (binary) file transfer mode is intended for transferring compiled files (executables). The ascii file transfer mode is intended for transferring text files, such as config files. The default selection of bin mode should be sufficient in all cases when copying files to a remote FTP server. • **sftp://**[*username*@]*server***/***path*[**/***filename*]—Specifies the SFTP network server and, optionally, the renamed file. • **tftp://***server*[**:***port*]**/***path*[**/***filename*]—Specifies the TFTP network server and, optionally, the renamed file. When you select a destination file system using **ftp:**, **sftp:**, or **tftp:**, the ACE performs the following tasks: • Prompts you for your username and password if the destination file system requires user authentication. • Prompts you for the server information if you do not provide the information with the command. • Copies the file to the root directory of the destination file system if you do not provide path information.

Copying Files to a Remote Server

This section describes how to copy a file from Flash memory on the ACE to a remote server using FTP, SFTP, or TFTP. The copy serves as a backup file for such files as the capture buffer file, core dump, ACE licenses in .tar format, running-configuration file, or startup-configuration file.

Detailed Steps

Command	Purpose
copy {**core:***filename* | **disk0:**[*path/*]*filename* | **running-config** | **startup-config**} {**ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} **Example:** host1/Admin# copy running-config ftp://192.168.215.124/running-config_Adminctx Enter username[]? user1 Enter the file transfer mode[bin/ascii]: [bin] Password: password1 Passive mode on. Hash mark printing on (1024 bytes/hash mark). ####	Copies a file from Flash memory on the ACE to a remote server using FTP, SFTP, or TFTP. The keywords, arguments, and options are as follows: • **core:***filename*—Specifies a core dump residing on the ACE in Flash memory (see the "Managing Core Dump Files" section). The **copy core:** command is available only in the Admin context. Use the **dir core:** command to view the core dump files available in the core: file system. Copy the complete filename (for example, 0x401_vsh_log.25256.tar.gz) by using the **copy core:** command. • **disk0:**[*path/*]*filename***—**Specifies a file in the disk0: file system of Flash memory (for example, a packet capture buffer file, ACE licenses in .tar format, or a system message log). Use the **dir disk0:** command to view the files available in the disk0: file system. • **running-config**—Specifies the running-configuration file residing on the ACE in volatile memory. • **startup-config**—Specifies the startup-configuration file currently residing on the ACE in Flash memory. • **ftp://***server/path*[*/filename*]—Specifies the FTP network server and, optionally, the renamed file. When using FTP, the bin (binary) file transfer mode is intended for transferring compiled files (executables). The ascii file transfer mode is intended for transferring text files, such as config files. The default selection of bin mode should be sufficient in all cases when copying files to a remote FTP server. • **sftp://**[*username@*]*server/path*[*/filename*]—Specifies the SFTP network server and, optionally, the renamed file. • **tftp://***server*[**:***port*]**/***path*[*/filename*]—Specifies the TFTP network server and, optionally, the renamed file. When you select a destination file system using **ftp:**, **sftp:**, or **tftp:**, the ACE performs the following tasks: • Prompts you for your username and password if the destination file system requires user authentication. • Prompts you for the server information if you do not provide the information with the command. • Copies the file to the root directory of the destination file system if you do not provide path information.

Copying Files from a Remote Server

This section describes how to copy a file from a remote server to a location on the ACE using FTP, SFTP, or TFTP.

Detailed Steps

Command	Purpose
copy {**ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} {**disk0:**[*path/*]*filename* | **image:***image_name* | **running-config** | **startup-config**} **Example:** host1/Admin# copy ftp://192.168.1.2/ startup-config Enter source filename[]? startup_config_Adminctx File already exists, do you want to overwrite?[y/n]: [y] y Enter username[]? user1 Enter the file transfer mode[bin/ascii]: [bin] Password: Passive mode on. Hash mark printing on (1024 bytes/hash mark).	Copies a file from a remote server to a location on the ACE using FTP, SFTP, or TFTP. The keywords, arguments, and options are as follows: • **ftp://***server/path*[*/filename*]—Specifies the FTP network server and, optionally, the filename. When using FTP, the bin (binary) file transfer mode is intended for transferring compiled files (executables). The ascii file transfer mode is intended for transferring text files, such as config files. The default selection of bin mode should be sufficient in all cases when copying files to a remote FTP server. • **sftp://**[*username@*]*server/path*[*/filename*]—Specifies the SFTP network server and, optionally, the filename. • **tftp://***server*[**:***port*]*/path*[*/filename*]—Specifies the TFTP network server and, optionally, the filename. • **disk0:**[*path/*]*filename***—**Specifies a file destination in the disk0: file system of Flash memory. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system. • **image:***image_name*—Specifies to copy a system software image to Flash memory. Use the **boot system** command as described in Chapter 1, Setting Up the ACE to specify the BOOT environment variable. The BOOT environment variable specifies a list of image files on various devices from which the ACE can boot at startup. • **running-config**—Specifies to replace the running-configuration file currently residing on the ACE in RAM (volatile memory). • **startup-config**—Specifies to replace the startup-configuration file currently residing on the ACE in Flash memory (nonvolatile memory).

Copying an ACE Software System Image to a Remote Server

This section describes how to copy an ACE software system image from Flash memory to a remote server using FTP, SFTP, or TFTP.

Restrictions

The **copy image:** command is available in the Admin context only.

Detailed Steps

	Command	Purpose
Step 1	**dir image:** **Example:** host1/Admin# dir image:	(Optional) Displays the software system images available in Flash memory.
Step 2	**show version** **Example:** host1/Admin# show version	(Optional) Displays the version information of system software that is loaded in flash memory and currently running on the ACE.
Step 3	**copy image:***filename* {**ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} **Example:** host1/Admin# copy image:sb-ace.NOV_11 ftp://192.168.1.2	Copies an ACE software system image from Flash memory to a remote server using FTP, SFTP, or TFTP. The keywords, arguments, and options are as follows: • *filename*—Name of the ACE system software image. • **ftp://***server***/***path*[**/***filename*]—Specifies the FTP network server and, optionally, the renamed software system image. • **sftp://**[*username***@**]*server***/***path*[**/***filename*]—Specifies the SFTP network server and, optionally, the renamed software system image. • **tftp://***server*[**:***port*]**/***path*[**/***filename*]—Specifies the TFTP network server and, optionally, the renamed software system image. When you select a destination file system using **ftp:**, **sftp:**, or **tftp:**, the ACE performs the following tasks: • Prompts you for your username and password if the destination file system requires user authentication. • Prompts you for the server information if you do not provide the information with the command. • Copies the file to the root directory of the destination file system if you do not provide path information.

Uncompressing Files in the disk0: File System

This section describes how to uncompress (unzip) LZ77 coded files in the disk0: file system (for example, zipped probe script files).

Restrictions

The filename must end with a .gz extension for the file to be uncompressed using the **gunzip** command. The .gz extension indicates a file zipped by the gzip (GNU zip) compression utility.

Detailed Steps

	Command	Purpose
Step 1	**dir disk0:**[*directory/*][*path/*][*filename*] **Example:** host1/Admin# dir disk0:	(Optional) Displays a list of available zipped files on the disk0: file system. The arguments are as follows: • *directory/*—(Optional) Contents of the specified directory. • *path/*—(Optional) Path to display the contents of a specific directory on the disk0: file system. • *filename*—(Optional) Information that relates to the specified file, such as the file size and the date it was created. You can use wildcards in the filename. A wildcard character (*) matches all patterns. Strings after a wildcard are ignored.
Step 2	**gunzip disk0:***filename* **Example:** host1/Admin# gunzip disk0:PROBE_SCRIPTS.gz	Uncompresses (unzips) LZ77 coded files in the disk0: file system. The *filename* argument identifies the name of the compressed file on the disk0: file system. The filename must end with a .gz extension.

Untarring Files in the disk0: File System

This section describes how to untar a single file with a .tar extension in the disk0: file system. Use this process to untar the sample scripts file or to unzip a back-up licenses created with the **copy licenses disk0:** command if a license becomes corrupted or lost.

A .tar file keeps related files together and facilitates the transfer of multiple files. A .tar file is a series of separate files, typically not compressed, added together into a single file by a UNIX TAR program. The resulting file is known as a tarball, which is similar to a ZIP file but without the compression. The files in a .tar file must be extracted before they can be used.

Restrictions

To untar a file, the filename must end with a .tar extension.

Detailed Steps

Command	Purpose
untar disk0:[*path/*]*filename* **Example:** host1/Admin# untar disk0:mylicenses.tar	Untars a single file with a .tar extension in the disk0: file system. The *filename* argument identifies the name of the .tar file in the disk0: file system. You can optionally provide a path to the .tar file if it exists in another directory in the disk0: file system.

Creating a New Directory

This section describes how to create a directory in the disk0: file system of Flash memory.

Detailed Steps

Command	Purpose
mkdir disk0:[*path/*]*directory* **Example:** host1/Admin# mkdir disk0:TEST_DIRECTORY	Create a directory in the disk0: file system of Flash memory. The arguments are as follows: • *path/*—(Optional) Path on the disk0: file system to the new directory. Specify the optimal path if you want to create a directory within an existing directory. • *directory*—Name of the directory to create in disk0:. If a directory with the same name already exists, the ACE does not create the new directory and the "Directory already exists" message appears.

Deleting an Existing Directory

This section describes how to remove an existing directory from the disk0: file system of Flash memory.

Prerequisites

The directory must be empty before you can delete it. To remove a file from the ACE file system, use the **delete** command (see the "Deleting Files" section).

Detailed Steps

	Command	Purpose
Step 1	**dir disk0:** **Example:** host1/Admin# dir disk0:	(Optional) Displays the contents of the disk0: file system.
Step 2	**rmdir disk0:**[*path/*]*directory* **Example:** host1/Admin# rmdir disk0:TEST_DIRECTORY	Removes an existing directory from the disk0: file system of Flash memory. The *directory* argument provides the name of the directory to delete from the disk0: file system. The directory must be empty before you can delete it. You can optionally provide a path to a directory in the disk0: file system.

Moving Files

This section describes how to move a file between directories in the disk0: file system. If a file with the same name already exists in the destination directory, that file is overwritten by the moved file.

Detailed Steps

	Command	Purpose
Step 1	**dir disk0:** **Example:** host1/Admin# dir disk0:	(Optional) Displays the files available in the disk0: file system.
Step 2	**move disk0:**[*source_directory/*]*filename* **disk0:**[*destination_directory/*]*filename* **Example:** host1/Admin# move disk0:SAMPLEFILE disk0:MYSTORAGE/SAMPLEFILE	Moves a file between directories in the disk0: file system. The keywords and arguments are as follows: • *source_directory*—(Optional) Name of the source directory in the disk0: file system. • *destination_directory*—(Optional) Name of the destination directory in the disk0: file system. • *filename*—Name of the file to move in the disk0: file system.

Deleting Files

This section describes how to delete a file from a specific file system in the ACE. When you delete a file, the ACE erases the file from the specified file system.

Note To remove a directory from the ACE file system, use the **rmdir** command (see the "Deleting an Existing Directory" section).

Detailed Steps

	Command	Purpose
Step 1	**dir** {**core:** | **disk0:** | **image:** | **volatile:**} **Example:** host1/Admin# dir disk0:	(Optional) Displays the files available in the specified file system.
Step 2	**delete** {**core:***filename* | **disk0:**[*directory/*]*filename* | **image:***filename* | **volatile:***filename*} **Example:** host1/Admin# delete disk0:mystorage/my_running-config1	Delete a file from a specific file system in the ACE. The keywords and arguments are as follows: • **core:***filename*—Deletes the specified file from the core: file system (see the "Managing Core Dump Files" section). The **delete cores:** command is available only in the Admin context. • **disk0:**[*directory/*]*filename*— Deletes the specified file from the disk0: file system (for example, a packet capture buffer file or system message log). You can optionally provide a path to a file in directory in the disk0: file system. • **image:***filename*—Deletes the specified file from the image: file system. The **delete image:** command is available only in the Admin context. • **volatile:***filename*—Deletes the specified file from the volatile: file system.

Displaying Files Residing On the ACE

To display the files in a directory and the contents of a file, perform the following tasks:

Command	Purpose
dir {**core:** | **disk0:**[*directory/*][*filename*] | **image:**[*filename*] | **probe:**[*filename*] | **volatile:**[*filename*]}	Displays a detailed list of directories and files contained within the specified file system on the ACE, including names, sizes, and time created. The keywords and arguments are as follows: • **core:**—Displays the contents of the core: file system. • **disk0:**—Displays the contents of the disk0: file system. • **image:**—Displays the contents of the image: file system. • **probe:**—Displays the contents of the probe: file system. This directory contains the Cisco-supplied scripts. For more information about these scripts, see the *Cisco Application Control Engine Module Server Load-Balancing Configuration Guide*. • **volatile:**—Displays the contents of the volatile: file system. • *directory/*—(Optional) Contents of the specified directory. • *filename*—(Optional) Information that relates to the specified file, such as the file size and the date it was created. You can use wildcards in the filename. A wildcard character (*) matches all patterns. Strings after a wildcard are ignored.
show file {**disk0:** [**path/**]*filename* | **volatile:** *filename*} [**cksum** | **md5sum**]	Displays the contents of a specified file in a directory in Flash memory or in nonvolatile memory. The keywords, arguments, and options are as follows: • **disk0:** [*path/*]*filename*—Specifies the name of a file residing in the disk0: file system of Flash memory (for example, a packet capture buffer file or system message log). You can optionally provide a path to a file in a directory in the disk0: file system. • **volatile:** *filename***—**Specifies the name of a file in the volatile memory file system of the ACE. • **cksum**—(Optional) Displays the cyclic redundancy check (CRC) checksum for the file. The checksum values compute a CRC for each named file. Use this option to verify that the file is not corrupt. You compare the checksum output for the received file against the checksum output for the original file. • **md5sum**—(Optional) Displays the MD5 checksum for the file. MD5 is an electronic fingerprint for the file. MD5 is the latest implementation of the internet standards described in RFC 1321 and is useful for data security and integrity.

Examples

The following example shows the output of the **dir disk0:** commands:

```
host1/Admin# dir disk0:

     7465  Jan 03 00:13:22 2000 C2_dsb
     2218  Mar 07 18:38:03 2006 ECHO_PROBE_SCRIPT4
     1024  Feb 16 12:47:24 2006 core_copies_dsb/
     1024  Jan 01 00:02:07 2000 cv/
```

```
  1024  Mar 13 13:53:08 2006 dsb_dir/
    12  Jan 30 17:54:26 2006 messages
  7843  Mar 09 22:19:56 2006 running-config
  4320  Jan 05 14:37:52 2000 startup-config
  1024  Jan 01 00:02:28 2000 www/

        Usage for disk0: filesystem
               4254720 bytes total used
               6909952 bytes free
              11164672 bytes total
```

For example, to list the core dump files in Flash memory, enter:

```
host1/Admin# dir core:

    2261  Jan 13 18:33:02 2010 SYSTEM_STATS
  437478  Apr 15 13:40:36 2010 0x201_vsh_log.29732.tar.gz
  504105  Apr 21 20:23:45 2010 0x201_vsh_log.6957.tar.gz
  500547  Apr 24 10:58:26 2010 0x201_vsh_log.6959.tar.gz

        Usage for core: filesystem
                 2524160 bytes total used
               200572928 bytes free
               203097088 bytes total
```

Saving show Command Output to a File

This section describes how to save all **show** screen output to a file by appending **>** *filename* to any command. For example, you can enter **show interface >** *filename* at the Exec mode CLI prompt to redirect the interface configuration command output to a file created at the same directory level.

Detailed Steps

Command	Purpose
show *keyword* [**|** {**begin** *pattern* **|** **count** **|** **end** **|** **exclude** *pattern* **|** **include** *pattern* **|** **last** **|** **more**}] [**>** {*filename* **|** {**disk0:|** **volatile**}:[*path/*][*filename*] **|** {**ftp:**//*server/path*[*/filename*] **|** **sftp:**//[*username@*]*server/path*[*/filename*] **|** **tftp:**//*server*[*:port*]*/path*[*/filename*]} **Example:** host1/Admin# show running-config > ftp://192.168.1.2	Saves a **show** command output to a file. The arguments, keywords, and options are as follows: • **|**—(Optional) Enables an output modifier that filters the command output. • **begin** *pattern*—Begins with the line that matches the pattern that you specify. • **count**—Counts the number of lines in the output. • **end** *pattern*—Ends with the line that matches the pattern that you specify. • **exclude** *pattern*—Excludes the lines that match the pattern that you specify. • **include** *pattern*—Includes the lines that match the pattern that you specify. • **last**—Displays the last few lines of the output. • **more**—Displays one window page at a time. • **>**—(Optional) Enables an output modifier that redirects the command output to a file. • *filename*—Name of the file that the ACE saves the output to on the volatile: file system. • **disk0:**—Specifies that the destination is the disk0: file system on the ACE Flash memory. • **volatile:**—Specifies that the destination is the volatile: file system on the ACE. • [*path/*][*filename*]—(Optional) Path and filename to the disk0: or volatile: file system. • **ftp://***server/path*[*/filename*]—Specifies the FTP network server and, optionally, a filename. • **sftp://**[*username@*]*server/path*[*/filename*]—Specifies the SFTP network server and, optionally, a filename. • **tftp://***server*[**:***port*]*/path*[*/filename*]—Specifies the TFTP network server and, optionally, a filename.

Using Backup and Restore

This section describes how to back up and restore your ACE module configuration data and dependent files. It contains the following subsections:

- Information About the Backup and Restore Features
- Guidelines and Limitations
- Defaults

Information About the Backup and Restore Features

This section provides information about the backup and restore features. With these features, you can back up or restore the configuration and dependencies of an entire ACE or of a particular virtual context. Configuration dependencies are those files that are required to exist on the ACE so that a configuration can be applied to it. Such files include health-monitoring scripts, SSL certificates, SSL keys, and so on.

Note The ACE backs up the dependencies that exist at the time when the backup is performed.

This feature allows you to back up and restore the following configuration files and dependencies:

- Running-configuration files
- Startup-configuration files
- Checkpoints
- SSL certificates
- SSL keys
- Health-monitoring scripts
- Licenses

Note The backup feature does not back up the sample SSL certificate and key pair files.

Typical uses for this feature are as follows:

- Back up a configuration for later use
- Recover a configuration that was lost because of a software failure or user error
- Restore configuration files to a new ACE when a hardware failure resulted in an RMA of the old ACE
- Transfer the configuration files to a different ACE

The **backup** and **restore** commands are supported in both the Admin and user contexts. If you enter these commands in the Admin context, you can back up or restore the configuration files for either the Admin context only or for all contexts in the ACE. If you enter the commands in a user context, you can back up or restore the configuration files only for that context.

Both the **backup** and the **restore** commands run asynchronously (in the background). You can monitor their progress by entering their corresponding **show** commands.

Archive File

The **backup** command runs asynchronously, that is, it runs in the background, which allows you to enter other commands at the CLI while the ACE processes the backup. When you instruct the ACE to back up the selected files, the ACE tars and GZIP-compresses them into a .tgz archive file and places the file in disk0:. For the Admin context, you can store one archive for the Admin context and one archive for the entire ACE. For a user context, you can store one archive for that context only. You can later use the archive files to restore the state of the same ACE or a different ACE.

Each time that you create a new backup for the entire ACE or for a particular user context, the ACE overwrites the previous ACE-wide archive or the context-specific archive, respectively.

Archive Naming Conventions

Archive files for individual contexts have the following naming convention format:

Hostname_ctxname_timestamp.tgz

where *timestamp* has the following format: *yyyy_mm_dd_hh_mm_ss*

For example:

```
ACE-1_ctx1_2009_08_30_15_45_17.tgz
```

If you back up the entire ACE, the archive filename does not include the *ctxname* field. So, the format is as follows:

Hostname_timestamp.tgz

For example:

```
ACE-1_2009_08_30_15_45_17.tgz
```

Archive Directory Structure and Filenames

The ACE uses a flat directory structure for the backup archive. The ACE provides file extensions for the individual files that it backs up so that you can identify the types of files easily when restoring an archive. All files are stored in a single directory that is tarred and GZIPed as follows:

```
ACE-1_Ctx1_2009_08_30_15_45_17.tgz
 ACE-1_Ctx1_2009_08_30_15_45_17\
  context_name-running
  context_name-startup
  context_name-chkpt_name.chkpt
  context_name-cert_name.cert
  context_name-key_name.key
  context_name-script_name.tcl
  context_name-license_name.lic
```

When you choose to encrypt the key pair files in a backup archive, the ACE appends an .enc extension to the filename (*context_name-key_name*.enc).

Guidelines and Limitations

The backup and restore features have the following configuration guidelines and limitations:

- Use the Admin context for an ACE-wide backup and the corresponding context for a user context backup.

- When you back up the running-configuration file, the ACE uses the output of the **show running-configuration** command as the basis for the archive file.
- The ACE backs up only exportable certificates and keys.
- License files are backed up only when you back up the Admin context.
- Use a passphrase to back up SSL keys in encrypted form. Remember the passphrase or write it down and store it in a safe location. When you restore the encrypted keys, you must enter the passphrase to decrypt the keys. If you use a passphrase when you back up the SSL keys, the ACE encrypts the keys with AES-256 encryption using OpenSSL software.
- Only probe scripts that reside in disk0: need to be backed up. The prepackaged probe scripts in the probe: directory are always available. When you perform a backup, the ACE automatically identifies and backs up the scripts in disk0: that are required by the configuration.
- The ACE does not resolve any other dependencies required by the configuration during a backup except for scripts that reside in disk0:. For example, if you configured SSL certificates in an SSL proxy in the running-configuration file, but you later deleted the certificates, the backup proceeds as if the certificates still existed.
- To perform a backup or a restore operation, you must have the admin RBAC feature in your user role.
- When you instruct the ACE to restore the archive for the entire ACE in the Admin context, it restores the Admin context completely first, and then it restores the other contexts. The ACE restores all dependencies before it restores the running context. The order in which the ACE restores dependencies is as follows:
  - License files
  - SSL certificates and key files
  - Health-monitoring scripts
  - Checkpoints
  - Startup-configuration file
  - Running-configuration file
- After you restore license files, previously installed license files are uninstalled and the restored files are installed in their place.
- In a redundant configuration, if the archive that you want to restore is different from the peer configurations in the FT group, redundancy may not operate properly after the restoration.
- You can restore a single context from an ACE-wide backup archive provided that:
  - You enter the **restore** command in the context that you want to restore
  - All files dependencies for the context exist in the ACE-wide backup archive

Defaults

Table 4-2 lists the default settings for the backup and restore feature parameters.

Table 4-2 Default Backup and Restore Parameters

Parameter	Default
Backed up files	By default, the ACE backs up the following files in the current context: • Running-configuration file • Startup-configuration file • Checkpoints • SSL certificates • SSL keys • Health-monitoring scripts • Licenses
SSL key backup encryption	None

Backing Up the ACE Configuration Files and Dependencies

This section describes the procedure that you perform to back up the ACE configuration files and dependencies.

Restrictions

To back up all contexts, you must be in the Admin context and you must specify the **all** keyword.

Detailed Steps

	Command	Purpose
Step 1	**changeto** **Example:** host1/Admin# changeto C1 host1/C1#	Changes to the specified context. Be sure that you are in the context that you wish to back up. To back up all contexts in the ACE, you must be in the Admin context.
Step 2	**backup** [**all**][**pass-phrase** *text_string*][**exclude** *component*] **Example:** host1/Admin# backup all pass-phrase my_pass_phrase exclude checkpoints host1/Admin#	Backs up configuration files and dependencies in the current context or in all contexts in the ACE. The keywords and arguments of this command are as follows: • **all**—(Optional) Specifies that the ACE should back up the configuration files and dependencies in all contexts. You can specify this keyword only in the Admin context. • **pass-phrase** *text_string*—(Optional) Passphrase that you specify to encrypt the backed up SSL certificates or keys. Enter the passphrase as an unquoted text string with no spaces and a maximum of 40 alphanumeric characters. You must enter the **pass-phrase** keyword before the **exclude** keyword. If you enter a passphrase and then exclude the SSL files from the archive, the ACE does not use the passphrase. **Note** If you imported SSL certificates or keys with a crypto passphrase, you must use the **pass-phrase** option to encrpyt the crypto passphrase when you back up these files. • **exclude** *component*—(Optional) Specifies the components that you do not wish to back up. You can enter any of the following components in any order separated by a comma if you enter more than one: – **checkpoints**—Excludes all checkpoints – **ssl-files**—Excludes SSL certificate files and key files
Step 3	**show backup status** [**detail**] **Example:** host1/Admin# show backup status detail	(Optional) Displays the progress of the backup process for each component in the different ACE contexts. Use the **detail** option to view the components or files that have already been backed up in each context. When the backup is finished, the command displays the status as SUCCESS.
Step 4	**show backup errors** **Example:** host1/Admin# show backup errors	(Optional) If the backup fails, displays the errors that occurred during the backup process.

Restoring the ACE Configuration Files and Dependencies

This section describes the procedure that you perform to restore the ACE configuration files and dependencies on the same or a different ACE. Be sure that the backup archive file resides in disk0: prior to starting the restoration.

Caution The **restore** command clears any existing SSL certificate and key-pair files, license files, and checkpoints in a context before it restores the backup archive file. If your configuration includes SSL files or checkpoints and you excluded them when you created the backup archive, those files will no longer exist in the context after you restore the backup archive. To preserve any existing exportable SSL certificate and key files in the context, before you enter the **restore** command, export the certificates and keys that you want to keep to an FTP, SFTP, or TFTP server by using the **crypto export** command. After you restore the archive, import the SSL files into the context. For details on exporting and importing SSL certificate and key pair files, see the *Cisco Application Control Engine Module SSL Configuration Guide*.

You can also use the **exclude** option of the **restore** command to instruct the ACE not to clear the SSL files in disk0: and to ignore the SSL files in the backup archive when the ACE restores the backup.

Prerequisites

- The backup archive must reside in disk0: in the ACE where you want to restore the archive before you start the restoration.
- No automatic rollback will be done in case of a restore failure. We recommend that you back up the ACE before you attempt to restore an archive.
- If you excluded the SSL files from the backup archive, you must import the certificates and keys from the FTP, SFTP, or TFTP server before you restore the archive. Then, when you enter the **restore** command, enter the **exclude ssl-files** option.

Restrictions

You must be in the Admin context to restore all contexts.

Detailed Steps for a Nonredundant Configuration

Note This procedure will cause an interruption in service for the current context or for all contexts, depending on the type of backup archive that you are restoring. We recommend that you schedule the restoration of a backup archive on an ACE during a maintenance window.

	Command	Purpose
Step 1	**changeto** **Example:** host1/Acmin# changeto C1 host1/C1#	Changes to the specified context. Be sure that you are in the context in which you wish to restore the backup archive. To restore an ACE-wide backup archive completely, you must be in the Admin context.
Step 2	**restore** {[**all**] **disk0:***archive_filename*} [**pass-phrase** *text_string*] [**exclude** {**licenses** | **ssl-files**}] **Example:** host1/Admin# restore disk0:switch_Admin_07_July_2009_11_08_04_AM.tgz pass-phrase MY_PASS_PHRASE	Restores configuration files and dependencies in the current context or in all contexts in the ACE. The keywords and arguments of this command are as follows: • **all**—(Optional) Specifies that the ACE should restore the configuration files and dependencies in all contexts. You can specify this keyword only in the Admin context. • **disk0:***archive_filename*—Name of the archive file that you want to restore. • **pass-phrase** *text_string*—(Optional) Passphrase that you used to encrypt the backed up SSL keys in the archive. You must enter the **pass-phrase** option before you enter the **exclude** option. Enter the passphrase as an unquoted text string with no spaces and a maximum of 40 alphanumeric characters. If you used a passphrase when you backed up the SSL keys, the ACE encrypted the keys with AES-256 encryption using OpenSSL software. To restore the SSL keys, you must enter that same passphrase. **Note** If you forget your passphrase, import the required SSL files first. Then, use the **exclude** option of the **restore** command to restor e the archive. • **exclude**—(Optional) Instructs the ACE not to restore the following specified files: – **licenses**—Excludes license files from the restoration. Use this option when you want to keep the license files that are already installed in the ACE and ignore the license files in the backup archive, if any. **Note** If you upgrade to software version A4(1.0) or later from a release before A4(1.0), the ACE cannot install the earlier license files because they are unsupported. The ACE ignores these license files and keeps the existing licenses. – **ssl-files**—Excludes SSL certificates and keys from the restoration. Use this option only if you want to keep the SSL files already present in your ACE and ignore the SSL files in the backup archive, if any. **Note** If you enter the **exclude** option first, you cannot enter the **pass-phrase** option.

	Command	Purpose
Step 3	**show restore status** [**detail**] **Example:** host1/Admin# show restore status	(Optional) Displays the progress of the restore process by displaying the context. Use the **detail** option to view the components or files that have already been backed up in each context. When the restore is finished, the command displays the status as SUCCESS.
Step 4	**show restore errors** **Example:** host1/Admin# show restore errors	(Optional) If the restore fails, displays the errors that occurred during the restore process.

Detailed Steps to Restore a Redundant Configuration

Note This procedure will cause an interruption in service for the two redundant contexts. We recommend that you schedule the restoration of a backup archive on a redundant pair during a maintenance window.

	Command	Purpose
Step 1	**changeto** **Example:** host1/Admin# changeto C1 host1/C1#	Changes to the specified context. Be sure that you are in the context in which you wish to restore the backup archive. To restore an ACE-wide backup archive completely, you must be in the Admin context.
Step 2	**config** **Example:** host1/Admin# config host1/Admin(config)#	Enters configuration mode on the active member of the FT group.
Step 3	**ft group** *group_id* **no inservice** **Example:** host1/Admin(config)# ft group 1 host1/Admin(config-ft-group)# no inservice	Disables redundancy for the members of the FT group. You must take the FT group out of service before you can restore the archive on the standby ACE. Otherwise, configuration mode is disabled on the standby ACE and the restoration will fail with the following error message: Archive restore not allowed when config mode is disabled.
Step 4	**Ctrl-Z** **Example:** host1/Admin(config-ft-group)# Ctrl-Z host1/Admin#	Returns to Exec mode from any configuration mode.

	Command	Purpose
Step 5	**restore** {[**all**] **disk0:***archive_filename*} [**pass-phrase** *text_string*] [**exclude** {**licenses** | **ssl-files**}] **Example:** host1/Admin# restore disk0:switch_Admin_07_July_2009_11_08_04_AM.tgz pass-phrase MY_PASS_PHRASE	Restores configuration files and dependencies in the current context or in all contexts in the ACE. The keywords and arguments of this command are as follows: • **all**—(Optional) Specifies that the ACE should restore the configuration files and dependencies in all contexts. You can specify this keyword only in the Admin context. • **disk0:***archive_filename*—Name of the archive file that you want to restore. • **pass-phrase** *text_string*—(Optional) Passphrase that you used to encrypt the backed up SSL keys in the archive. You must enter the pass phrase before you use the **exclude** option. Enter the passphrase as an unquoted text string with no spaces and a maximum of 40 alphanumeric characters. If you used a passphrase when you backed up the SSL keys, the ACE encrypted the keys with AES-256 encryption using OpenSSL software. To restore the SSL keys, you must enter that same passphrase. **Note** If you forget your passphrase, import the required SSL files first. Then, use the **exclude** option of the **restore** command to restor e the archive. • **exclude**—(Optional) Instructs the ACE not to restore the following specified files: – **licenses**—Excludes license files from the restoration. Use this option when you want to keep the license files that are already installed in the ACE and ignore the license files in the backup archive, if any. **Note** If you upgrade to software version A4(1.0) or later from a release before A4(1.0), the ACE cannot install the earlier license files because they are unsupported. The ACE ignores these license files and keeps the existing licenses. – **ssl-files**—Excludes SSL certificates and keys from the restoration. Use this option only if you want to keep the SSL files already present in your ACE and ignore the SSL files in the backup archive, if any.
Step 6	**config** **Example:** host1/Admin# config host1/Admin(config)#	Enters configuration mode on the active member of the FT group.

	Command	Purpose
Step 7	**ft group** *group_id* **inservice** **Example:** host1/Admin(config)# ft group 1 host1/Admin(config-ft-group)# inservice	Enables redundancy for both members of the FT group.
Step 8	**Ctrl-Z** **Example:** host1/Admin(config-ft-group)# Ctrl-Z host1/Admin#	Returns to Exec mode from any configuration mode.
Step 9	**show restore status** [**detail**] **Example:** host1/Admin# show restore status detail	(Optional) Displays the progress of the restoration. Use the **detail** option to view the components or files that have been restored in each context. When the restoration is finished, the command displays the status as SUCCESS.
Step 10	**show restore errors** **Example:** host1/Admin# show restore errors	(Optional) If the restoration fails, displays the errors that occurred during the restore process.

Copying a Backup Archive to a Server

This section describes the procdure that you perform to copy a backup archive from the ACE to an FTP or an SFTP server so that you can then restore the archive on a different ACE.

Restrictions

To use the **copy backup** command or the **copy backup-all** command, you must have Admin privileges in the context where you enter the command.

Detailed Steps

	Command	Purpose
Step 1	**changeto** **Example:** host1/Admin# changeto C1 host1/C1#	Changes to the specified context. Be sure that you are in the context from which you wish to copy the backup archive.
Step 2	**copy** {**backup** | **backup-all**} {**ftp:**[*//path*] | **sftp:**[*//path*}} **Example:** host1/Admin# config host1/Admin(config)#	Copies a single-context or an ACE-wide backup archive to an FTP or an SFTP server. The keywords of this command are as follows: • **backup**—Copies the last successful single-context backup archive to the specified FTP or SFTP server. This keyword is available in both the Admin context and user contexts. • **backup-all**—Copies the last successful ACE-wide (all contexts) backup archive to the specified FTP or SFTP server. This keyword is available only in the Admin context. • **ftp:**[//path] | **sftp:**[//path]—Specifies the FTP or SFTP server where you want to copy the backup archive and, optionally, the file path or URI. **Note** If you renamed or deleted the backup archive in a context, the **copy backup** command fails and the ACE displays an error message.

Examples

The following example shows how to copy a backup archive file to an SFTP server:

```
switch/Admin# copy backup sftp:
Enter Address for the sftp server[]? 10.25.25.11
Enter the destination filename[]? [switch_Admin_2009_08_22_02_48_49.tgz]
Enter username[]? root
Connecting to 10.25.25.11...
root@10.25.25.11's password:
sftp> Uploading /TN-HOME/Admin/switch_Admin_2009_08_22_02_48_49.tgz to
/root/switch_Admin_2009_08_22_02_48_49.tgz
/TN-HOME/Admin/switch_Admin_2009_08_22_02_48_ 100% 6737     0.0KB/s   00:00
```

Displaying the Status of the Backup Operation

To display the status of the backup operation, perform the following task:

Command	Purpose
show backup status [**detail**]	Displays the the status of the last backup operation. Backup status details are not stored across reboots. Possible values in the Status column are as follows: • SUCCESS—The component was successfully backed up • FAILED—The component failed to be backed up • N/A—The component (for example, a checkpoint or probe script) being backed up contains 0 files

Examples

The following example shows the output of the **show backup status** command:

```
hello/Admin# show backup status
Backup Archive: switch_Admin_2009_08_30_15_45_17.tgz
Type          : Context
Start Time    : Wed Aug 30 15:45:16 2009
Finished Time : Wed Aug 30 15:45:17 2009
Status        : In Progress
Current vc    : Admin
Completed     : 1/1
```

The following example shows the output of the **show backup status detail** command:

```
host1/Admin# show backup status detail

Backup Archive: switch_Admin_2009_08_30_15_45_17.tgz
Type          : Context
Start Time    : Wed Aug 30 15:45:16 2009
Finished Time : Wed Aug 30 15:45:17 2009
Status        : SUCCESS
Current vc    : Admin
Completed     : 1/1
-----------------------+---------------+--------------------------+------------
Context                 component       Time                       Status
-----------------------+---------------+--------------------------+------------

Admin                   Running-cfg     Wed Aug 30 15:45:17 2009   SUCCESS
Admin                   Startup-cfg     Wed Aug 30 15:45:17 2009   SUCCESS
Admin                   Checkpoints     Wed Aug 30 15:45:17 2009   SUCCESS
Admin                   Cert/Key        Wed Aug 30 15:45:17 2009   N/A
Admin                   License         Wed Aug 30 15:45:17 2009   SUCCESS
Admin                   Probe script    Wed Aug 30 15:45:17 2009   N/A
```

Displaying the Status of the Restoration

To display the status of the restoration, perform the following task:

Command	Purpose
show restore status [**detail**]	Displays the status of the last restoration. Restoration status details are not stored across reboots.

Examples

The following example shows the output of the **show restore status** command:

```
host1/Admin# show restore status
Backup Archive: switch_2009_08_30_15_45_17.tgz
Type          : Context
Start Time    : Wed Aug 30 16:45:16 2009
Finished Time : -
Status        : In Progress
Current vc    : Admin
Completed     : 0/1
```

The following example shows the output of the **show restore status detail** command:

```
host1/Admin# show restore status detail
Backup Archive: switch_2009_08_30_15_45_17.tgz
Type          : Context
Start Time    : Wed Aug 30 16:45:16 2009
Finished Time : -
Status        : In Progress
Current vc    : Admin
Completed     : 0/1
------------------------+---------------+--------------------------+------------
Context                 component       Time                       Status
------------------------+---------------+--------------------------+------------
Admin                   License         Wed Aug 30 16:45:16 2009   SUCCESS
Admin                   Cert/Key        Wed Aug 30 16:45:16 2009   SUCCESS
Admin                   Probe script    Wed Aug 30 16:45:16 2009   SUCCESS
Admin                   Checkpoints     Wed Aug 30 16:45:16 2009   SUCCESS
Admin                   Startup-cfg     Wed Aug 30 16:45:17 2009   In Progress
```

Displaying Backup and Restore Errors

To display the errors that may have occurred during a backup or restore operation that did not succeed, perform the following tasks:

Command	Purpose
show backup errors	Displays errors that occur during a backup operation. For information about backup system messages, see the *Cisco Application Control Engine Module System Message Guide*.
show restore errors	Displays errors that occur during a restore operation. For information about restore system messages, see the *Cisco Application Control Engine Module System Message Guide*.

Examples

The following example shows the output of the **show backup errors** command after a backup failed because of a disk copy failure for checkpoints:

```
host1/Admin# show backup errors

Context: Admin
Component: Checkpoint

Error Details:
Internal Error, checkpoint copy failed
```

The following example shows the output of the **show restore errors** command after a restore failed because the running-configuration file differences could not be applied:

```
host1/Admin# show restore errors

Context: Admin
Component: Running-cfg
Below diff could not be applied
--

--
ssh key rsa 4096 force
ssh key dsa 2048 force
ssh key rsa1 4096 force
--
```

The following example shows the output of the **show restore errors** command after a restore failed because a probe was not present in either disk0: or in the probe: directory.

```
host1/Admin# show backup errors
Context: Admin
Component: Probe scripts
Error Details:
Error, probe PROBE_1 not found in disk0: or probe:
```

Managing Core Dump Files

This section describes how to manage the ACE core dump files. A core dump occurs when the ACE experiences a fatal error. The ACE writes information about the fatal error to the core: file system in Flash memory before a switchover or reboot occurs. The core: file system is the storage location for all core files generated during a fatal error. Three minutes after the ACE reboots, the saved last core file is restored from the core: file system back to its original RAM location. This restoration is a background process and is not visible to the user.

This section contains the following topics:

Guidelines and Limitations

This topic includes the following restrictions:

- The core: file system is available from the Admin context only.
- Core dump information is for Cisco Technical Assistance Center (TAC) use only. If the ACE becomes unresponsive, you can view the dump information in the core through the **show cores** command. We recommend that you contact TAC for assistance in interpreting the information in the core dump.
- The time stamp on the restored last core file displays the time when the ACE booted up, not when the last core was actually dumped. To obtain the exact time of the last core dump, check the corresponding log file with the same process identifier (PID).

Copying Core Dumps

This section describes how to copy a core dump from the ACE to the disk0: file system or to a remote server. The ACE copies a single file based on the provided process identifier.

Restrictions

You must perform this task from the Admin context only.

Detailed Steps

	Command	Purpose
Step 1	**dir core:** **Example:** host1/Admin# dir core:	(Optional) Displays the list of available core files. You can copy the complete filename (for example, 0x401_vsh_log.25256.tar.gz) into the **copy core:** command.
Step 2	**copy core:***filename* {**disk0:**[*path/*][*filename*] | **ftp:***//server/path*[*/filename*] | **sftp:***//*[*username@*]*server/path*[*/filename*] | **tftp:***//server*[*:port*]*/path*[*/filename*]} **Example:** host1/Admin# copy core:0x401_vsh_log.8249.tar.gz ftp://192.168.1.2 Enter the destination filename[]? [0x401_vsh_log.8249.tar.gz] Enter username[]? user1 Enter the file transfer mode[bin/ascii]: [bin] Password: Passive mode on. Hash mark printing on (1024 bytes/hash mark).	Saves a core dump from the ACE to the disk0: file system or to a remote server. The keywords, arguments, and options are as follows: • *filename*—Core dump that resides on the ACE in Flash memory. • **disk0:**[*path/*][*filename*]—Specifies a file location for the core dump in the disk0: file system and a filename for the core. • **ftp://***server/path*[*/filename*]—Specifies the FTP network server and, optionally, the renamed core dump. When using FTP, the bin (binary) file transfer mode is intended for transferring compiled files (executables). The ascii file transfer mode is intended for transferring text files, such as config files. The default selection of bin mode should be sufficient in all cases when copying files to a remote FTP server. • **sftp://**[*username@*]*server/path*[*/filename*]—Specifies the SFTP network server and, optionally, the renamed core dump. • **tftp://***server*[**:***port*]*/path*[*/filename*]—Specifies the TFTP network server and, optionally, the renamed core dump. When you select a destination file system using **ftp:**, **sftp:**, or **tftp:**, the ACE performs the following tasks: • Prompts you for your username and password if the destination file system requires user authentication. • Prompts you for the server information if you do not provide the information with the command. • Copies the file to the root directory of the destination file system if you do not provide path information.

Clearing the Core Directory

This section describes how to clear out all of the core dumps stored in the core: file system.

Restrictions

You must perform this task from the Admin context only.

Detailed Steps

	Command	Purpose
Step 1	**dir core:** **Example:** host1/Admin# dir core:	(Optional) Displays the list of available core files.
Step 2	**clear cores** **Example:** host1/Admin# clear cores	Clears out all of the core dumps stored in the core: file system.

Deleting a Core Dump File

This section describes how to delete a core dump file from the core: file system in Flash memory.

Restrictions

You must perform this task from the Admin context only.

Detailed Steps

	Command	Purpose
Step 1	**dir core:** **Example:** host1/Admin# dir core:	(Optional) Displays the list of available core files. You can copy the complete filename (for example, 0x401_vsh_log.25256.tar.gz) into the **delete core:** command.
Step 2	**delete core:***filename* **Example:** host1/Admin# delete core:0x401_VSH_LOG.25256.TAR.GZ	Deletes a core dump file from the core: file system in Flash memory. The *filename* argument specifies the name of a core dump file located in the core: file system.

Capturing Packet Information

This section describes how to capture packet information, which is useful for troubleshooting connectivity problems with the ACE or for monitoring suspicious activity. The ACE can track packet information for network traffic that passes through the ACE. The attributes of the packet are defined by an ACL. The ACE buffers the captured packets, and you can copy the buffered contents to a file in Flash memory on the ACE or to a remote server. You can also display the captured packet information on your console or terminal.

Caution The packet capture function uses ACL resources as can be seen with the **show np 1 access-list resource** command. If you have a large ACL configuration and you enable packet capturing, the ACE may oversubscribe the allocated ACL resources. If this happens, you may see one of the following error messages:

In exec mode,

```
Error: Device Name:[0x3FF] Instance:[63] Error Type:[(null)] code:[255]
```

In config mode,

```
Error: ACL merge add acl to list failed
```

For information about using the **show np 1 access-list resource** command to monitor ACL resources and how to resolve ACL oversubscription problems, see the "Troubleshooting ACLs" section of the ACE Troubleshooting Wiki.

This section contains the following topics:

Enabling the Packet Capture Function

This section describes how to enable the packet capture function on the ACE for packet sniffing and network fault isolation. As part of the packet capture process, you specify whether to capture packets from all input interfaces or an individual VLAN interface. The packet capture feature streams output on the console as packets are received by the ACE.

Prerequisites

To create a capture based on an access list, the access list must already exist. For information about creating an access list, see the *Cisco Application Control Engine Module Security Configuration Guide*.

Restrictions

This topic includes the following restrictions:

- The packet capture function enables access-control lists (ACLs) to control which packets are captured by the ACE on the input interface. If the ACLs are selecting an excessive amount of traffic for the packet capture operation, the ACE will see a heavy load, which can cause a degradation in performance. We recommend that you avoid using the packet capture function when high network performance is critical.

  In addition, probe traffic will not hit a security ACL so ACLs cannot control the capture of those packets. In this case, probe traffic cannot be captured by the packet capture function.

- The capture packet function works on an individual context basis. The ACE traces only the packets that belong to the current context where you execute the **capture** Exec mode command. The context ID, which is passed along with the packet, can be used to isolate packets that belong to a specific context. To trace the packets for a specific context, use the **changeto** Exec mode command to enter the specified context and then use the **capture** command.
- If you enable packet capture for jumbo packets, the ACE captures only the first 1,860 bytes of data.
- The ACE does not automatically save the packet capture to a file. To copy the capture buffer information as a file in Flash memory or to a remote server, use the **copy capture** command (see the "Copying Packet Capture Buffer Information" section).

- When capturing packets based on a specific interface and you delete the interface, the ACE stops the capture automatically. If you check the status of the packet capture using the **show capture status** command, you will notice that the capture stopped because of an interface deletion. At this point, you can perform any operation (for example, saving the old capture) on the capture except starting the capture. To restart the capture, you must delete the old capture and configure a new one. The ACE handles the deletion of an ACL or an ACL entry in a similar manner.
- When capturing packets based on a specific access list name, ensure that the access list is for an input interface. If you configure the packet capture on the output interface, the ACE will fail to match any packets.
- If you add an interface while you are already capturing all interfaces, the capture continues using all the original interfaces. If you add an ACL entry during an existing ACL capture, the capture continues normally using the original ACL criteria.
- If the ACE stops a packet capture because of an interface or ACL deletion, the following additional information appears in the output of the **show capture** *buffer_name* **status** command:

```
Capture forced to stop due to change in [interface | access-list] config.
To restart the capture, remove and add the capture again.
```

- Under high traffic conditions, you may observe up to 64 packets printing on the console after you enter the **stop** keyword. These additional messages can occur because the packets were in transit or buffered before you entered the **stop** keyword.

Detailed Steps

Command	Purpose
capture *buffer_name* {{**all** | {**interface vlan** *number*}} **access-list** *name* [**bufsize** *buf_size* [**circular-buffer**]]} | **remove** | **start** | **stop** **Example:** host1/Admin# capture capture1 interface vlan50 access-list acl1 host1/Admin# capture capture1 start host1/Admin# capture capture1 stop	Enables the packet capture function on the ACE for packet sniffing and network fault isolation. The keywords, arguments, and options are as follows: • *buffer_name*—Name of the packet capture buffer. This argument associates the packet capture with a name. Specify a text string from 1 to 80 alphanumeric characters. • **all**—Specifies capture packets for all input interfaces. • **interface**—Specifies the interface from which to capture packets. • **vlan** *number*—Specifies the VLAN identifier associated with the specified input interface. • **access-list** *name*—Selects packets based on an existing access list. A packet must pass the access list filters before the packet is stored in the capture buffer. Specify a previously created access list identifier. Enter an unquoted text string with a maximum of 64 alphanumeric characters. • **bufsize** *buf_size*—(Optional) Specifies the buffer size, in kilobytes (KB), to store the packet capture. The range is from 1 to 5000 KB. The default is 64 KB. • **circular-buffer**—(Optional) Enables the packet capture buffer to overwrite itself, starting from the beginning, when the buffer is full. • **remove**—Removes the packet capture configuration. • **start**—Starts the packet capture function and displays the messages on the session console as the ACE receives the packets. The CLI prompt returns and you can type other commands at the same time that the ACE is capturing packets. To stop the capture process, enter **stop**. The packet capture function automatically stops when the buffer is full unless you enable the circular buffer function. • **stop**—Stops the packet capture process after a brief delay.

Copying Packet Capture Buffer Information

This section describes how to copy an existing packet capture buffer to the disk0: file system.

Detailed Steps

Command	Purpose
copy capture *capture_name* **disk0:** [*path/*]*destination_name* **Example:** host1/Admin# copy capture packet_capture_Jan_17_06 disk0: mycapture1	Copies an existing packet capture buffer to the disk0: file system The keywords, arguments, and options are as follows: • *capture_name*—Name of the packet capture buffer in Flash memory. Specify a text string from 1 to 80 alphanumeric characters. If necessary, use the **show capture** command to view the files available in Flash memory. This list includes the name of existing packet capture buffers. • **disk0:**—Specifies that the buffer is copied to the disk0: file system. Include a space between **disk0**: and a destination path. • [*path*/]*destination_name*—Destination path (optional) and name for the packet capture buffer. Specify a text string from 1 to 80 alphanumeric characters. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system.

Displaying or Clearing Packet Information

This section describes how to display or clear packet information and contains the following topics:

- Displaying Packet Information
- Clearing Capture Buffer Information

Displaying Packet Information

To display packet information, perform the following task:

Command	Purpose
show capture *buffer_name* [**detail** [**connid** *connection_id* | **range** *packet_start packet_end*] | **status**]	Displays the packet information that the ACE traces as part of the packet capture function. The keywords, arguments, and options are as follows: • *buffer_name*—Name of the packet capture buffer. Specify a text string from 1 to 80 alphanumeric characters. • **detail**—(Optional) Displays additional protocol information for each packet. • **connid** *connection_id*—(Optional) Displays protocol information for a specified connection identifier. • **range** *packet_start packet_end*—(Optional) Displays protocol information for a range of captured packets. • **status**—(Optional) Displays capture status information for each packet. For all types of received packets, the console display is in tcpdump format.

Clearing Capture Buffer Information

To clear the packet capture buffer, perform the following task:

Command	Purpose
clear capture *buffer_name*	Clears the capture packet buffer. The *buffer_name* argument specifies the name of the existing packet capture buffer to clear.

Using the Configuration Checkpoint and Rollback Service

This section describes how to make a checkpoint (or snapshot) of a running configuration on your ACE and how to use the rollback service to revert to the last known stable configuration.

At some point, you may want to modify your running configuration. If you run into a problem with the modified configuration, you may need to reboot your ACE. To prevent having to reboot your ACE after unsuccessfully modifying a running configuration, you can create a checkpoint (a snapshot in time) of a known stable running configuration before you begin to modify it. If you encounter a problem with the modifications to the running configuration, you can roll back the configuration to the previous stable configuration checkpoint.

The ACE allows you to make a checkpoint configuration at the context level. The ACE stores the checkpoint for each context in a hidden directory in Flash memory. If after you enter additional commands to modify the current running configuration, you enter the **rollback** command option, the ACE causes the running configuration to revert to the checkpointed configuration.

This section contains the following topics:

Creating a Configuration Checkpoint

This section describes how to create a configuration checkpoint.

Prerequisites

Be sure that the current running configuration is stable and is the configuration that you want to make a checkpoint.

Restrictions

This topic includes the following restrictions:

- The ACE supports a maximum of 10 checkpoints for each context.
- You must perform this task in the Exec mode of the context for which you want to create a checkpoint.
- Avoid using opening braces, closing braces, whitespaces, or any of the following symbols: `$&*()\|;'"<>/?

Detailed Steps

Command	Purpose

checkpoint create *name*
 	
Example:
`host1/Admin# checkpoint create MYCHECKPOINT`
`Generating configuration....`
`Created checkpoint 'MYCHECKPOINT'`
	Creates a configuration checkpoint.
	The *name* argument specifies the unique identifier of the checkpoint. Enter a text string with no spaces and a maximum of 25 alphanumeric characters.
	If the checkpoint already exists, the CLI responds with the following prompt:
	
	`Checkpoint already exists`
	`Do you want to overwrite it? (y/n) [n] y Generating configuration....`
	`Created checkpoint 'MYCHECKPOINT'`
	The default is **n**. If you do not want to overwrite the existing checkpoint, press **Enter**. To overwrite the existing checkpoint, enter **y**.

Deleting a Configuration Checkpoint

This section describes how to delete a configuration checkpoint.

Prerequisites

Before you use this command, make sure that you want to delete the checkpoint. When you enter this command, the ACE removes the checkpoint from Flash memory.

Detailed Steps

	Command	Purpose
Step 1	**show checkpoint all**	(Optional) Displays a list of all existing checkpoints.
Step 2	**checkpoint delete** *name* **Example:** `host1/Admin# checkpoint delete MYCHECKPOINT`	Deletes a configuration checkpoint. The *name* argument specifies the unique identifier of the checkpoint. Enter a text string with no spaces and a maximum of 25 alphanumeric characters.

Rolling Back a Running Configuration

This section describes how to roll back the current running configuration to the previously checkpointed running configuration for the current context.

Detailed Steps

	Command	Purpose
Step 1	**show checkpoint all** **Example:** host1/Admin# show checkpoint all	(Optional) Displays a list of all existing checkpoints.
Step 2	**show checkpoint detail** *name* **Example:** host1/Admin# show checkpoint MYCHECKPOINT5	(Optional) Displays the running configuration of the specified checkpoint.
Step 3	**checkpoint rollback** *name* **Example:** host1/Admin# checkpoint delete MYCHECKPOINT5 This operation will rollback the system's running configuration to the checkpoint's configuration. Do you wish to proceed? (y/n) [n] y Rollback in progress, please wait... Generating configuration.... Rollback succeeded host1/Admin#	Rolls back the current running configuration to the previously checkpointed running configuration for the current context. The *name* argument specifies the unique identifier of the checkpoint. Enter a text string with no spaces and a maximum of 25 alphanumeric characters.

If the running-configuration file has the **no ft auto-sync** command configured and the checkpoint has the **ft auto-sync** command configured, a checkpoint rollback will fail with the following message:

```
Warning : 'no ft auto-sync' & 'ft auto-sync' conflict detected - Rollback will fail
Failing Scenario - running config has 'no ft auto-sync' / checkpoint has 'ft auto-sync'
```

Copying a Checkpoint

This section desctibes how to copy a checkpoint to one of several destinations.

Detailed Steps

	Command	Purpose
Step 1	**show checkpoint all** **Example:** host1/Admin# show checkpoint all	(Optional) Displays a list of all existing checkpoints.
Step 2	**copy checkpoint:***filename* **disk0:**[*path/*]*filename* | **image:***image_name* | **startup-config** | {**ftp://***server/path*[**/***filename*] | **sftp://**[*username*]*server***/***path*[**/***filename*] | **tftp://***server*[**:***port*]**/***path*[**/***filename*]} **Example:** host1/Admin# **copy** **checkpoint:CHECKPOINT1.txt** **ftp://192.168.1.2** Enter the destination filename[]? [CHECKPOINT1.txt] Enter username[]? **user1** Enter the file transfer mode[bin/ascii]: [bin] Password: Passive mode on. Hash mark printing on (1024 bytes/hash mark). **Note** The **bin** (binary) file transfer mode is intended for transferring compiled files (executables). The **ascii** file transfer mode is intended for transferring text files, such as config files. The default selection of **bin** should be sufficient in all cases when copying files to a remote FTP server.	Copies the specificed checkpoint file to the specified destination. • *filename*—Filename of the checkpoint file residing on the ACE in flash memory. • **disk0:**[*path/*]*filename*—Specifies that the file destination is the disk0: directory of the current context and the filename for the checkpoint. If you do not provide the optional path, the ACE copies the file to the root directory on the disk0: file system. • **image:***image_name*—Specifies that the file destination is an image in the image: directory. • **startup-config**—Specifies that the destination file is the startup-configuration file. • **ftp://***server***/***path*[**/***filename*]—Specifies the File Transfer Protocol (FTP) network server and optional renamed checkpoint file. • **sftp://**[*username*@]*server***/***path*[**/***filename*]—Specifies the Secure File Transfer Protocol (SFTP) network server and optional renamed checkpoint file. • **tftp://***server*[**:***port*]**/***path*[**/***filename*]—Specifies the Trivial File Transfer Protocol (TFTP) network server and optional renamed checkpoint file.

Comparing a Checkpoint with the Running-Configuration File

This section describes how to compare a checkpoint with the running-configuration file.

Detailed Steps

	Command	Purpose
Step 1	**show checkpoint all** **Example:** `host1/Admin# show checkpoint all`	(Optional) Displays a list of all existing checkpoints.

Step 2	**compare checkpoint** *name*
	
	Example:
	`host1/Admin# compare checkpoint MYCHECKPOINT5`
	`Checkpoint config is same as running config`
	
	`host1/Admin#`
		Compares the specified checkpoint with the running-configuration file.
		
		The *name* argument specifies the unique identifier of an existing checkpoint. Enter a text string with no spaces and a maximum of 25 alphanumeric characters.
		
		If the checkpoint configuration is the same as the running-config, the output of this command is the following:
		
		`Checkpoint config is same as running config`
		
		If the checkpoint configuration is different from the running-config, the output will be the difference between the two configurations.

Displaying Checkpoint Information

To display checkpoint information, perform the following task:

Command	Purpose
show checkpoint {**all** | **detail** *name*} [|] [>]	Displays information relating to the configured checkpoints. • **all**—Displays a list of all existing checkpoints. The show output includes checkpoint time stamps. • **detail** *name*—Displays the running configuration of the specified checkpoint.

Table 4-3 describes the fields that appear in the **show checkpoint all** command output.

Table 4-3 Field Descriptions for the show checkpoint all Command Output

Field	Description
Checkpoint	Name of the checkpoint
Size	Size (in bytes) of the checkpoint
Date	Date and time at which the checkpoint was created

Reformatting the Flash Memory

The ACE uses the file allocation table (FAT16) as the base file system. The file system is used to allocate and organize storage space for various types of storage, such as startup-configuration files, SSL certificate storage, core files, image storage, and log files. Reformatting Flash memory on the ACE allows you to erase all data on the Flash memory and reformat it with the FAT16 version of the file allocation table. All user-defined configuration information is erased.

Caution We recommend that you reformat the ACE Flash memory only under the guidance and supervision of Cisco Technical Assistance Center (TAC).

Prerequisites

Before you reformat the Flash memory, we recommend that you copy the following ACE operation and configuration files or objects to a remote server:

- ACE software image
- ACE license
- Startup-configuration file of each context
- Running-configuration file of each context
- Core dump files of each context
- Packet capture buffers of each context
- SSL certificate and key pair files of each context

See the "Copying Files" section for details on how to use the **copy** command to save configuration files or objects, such as the existing startup-configuration files, running-configuration file, licenses, core dump files, or packet capture buffers, to a remote FTP, SFTP, or TFTP server.

See the *Cisco Application Control Engine Module SSL Configuration Guide* for details on how to use the **crypto export** command to export SSL certificate and key pair files to a remote FTP, SFTP, or TFTP server.

Detailed Steps

Command	Purpose
format disk0: **Example:** host1/Admin# format disk0: Warning!! This will reboot the system after formatting disk0. Do you wish to proceed anyway? (y/n) [n] **y**	Reformats Flash memory on the ACE and erases all data.

What to Do Next

After you reformat the Flash memory, perform the following actions:

- Reinstall the ACE software image by using the **copy image:** command (see the *Release Note for the Cisco Application Control Engine Module*).
- Reinstall the ACE license by using the **license install** command (see Chapter 3, Managing ACE Software Licenses).

- Import the startup and running-configuration files into the associated context by using the **copy** command (see the "Copying Configuration Files from a Remote Server" section).
- Import SSL certificate files and key pair files into the associated context using by the **crypto import** command (see the *Cisco Application Control Engine Module SSL Configuration Guide*).