



Using the Command-Line Interface

The command-line interface (CLI) is a line-oriented user interface that provides commands for configuring, managing, and monitoring the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following topics:

- [Accessing the ACE Command Modes](#)
- [Using the CLI Commands](#)
- [Getting CLI Help](#)
- [Creating Configuration Files Using a Text Editor](#)

**Note**

The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the ACE operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works or has the same function with the ACE.

Accessing the ACE Command Modes

When you log in to the ACE, you enter Exec mode. The Exec mode prompt begins with the hostname followed by the context name and pound sign (#). By default, the hostname for the ACE is switch. For example, if you log in to the Admin context, the following prompt appears:

```
switch/Admin#
```

If you log in as a user context (for example, you log in as user context C1), the following prompt appears:

```
switch/C1#
```

Exec mode has a set of commands that allow you to maintain the ACE and access configuration mode. To access configuration mode, use the **configure** command. This mode is identified by a (config) prompt. For example:

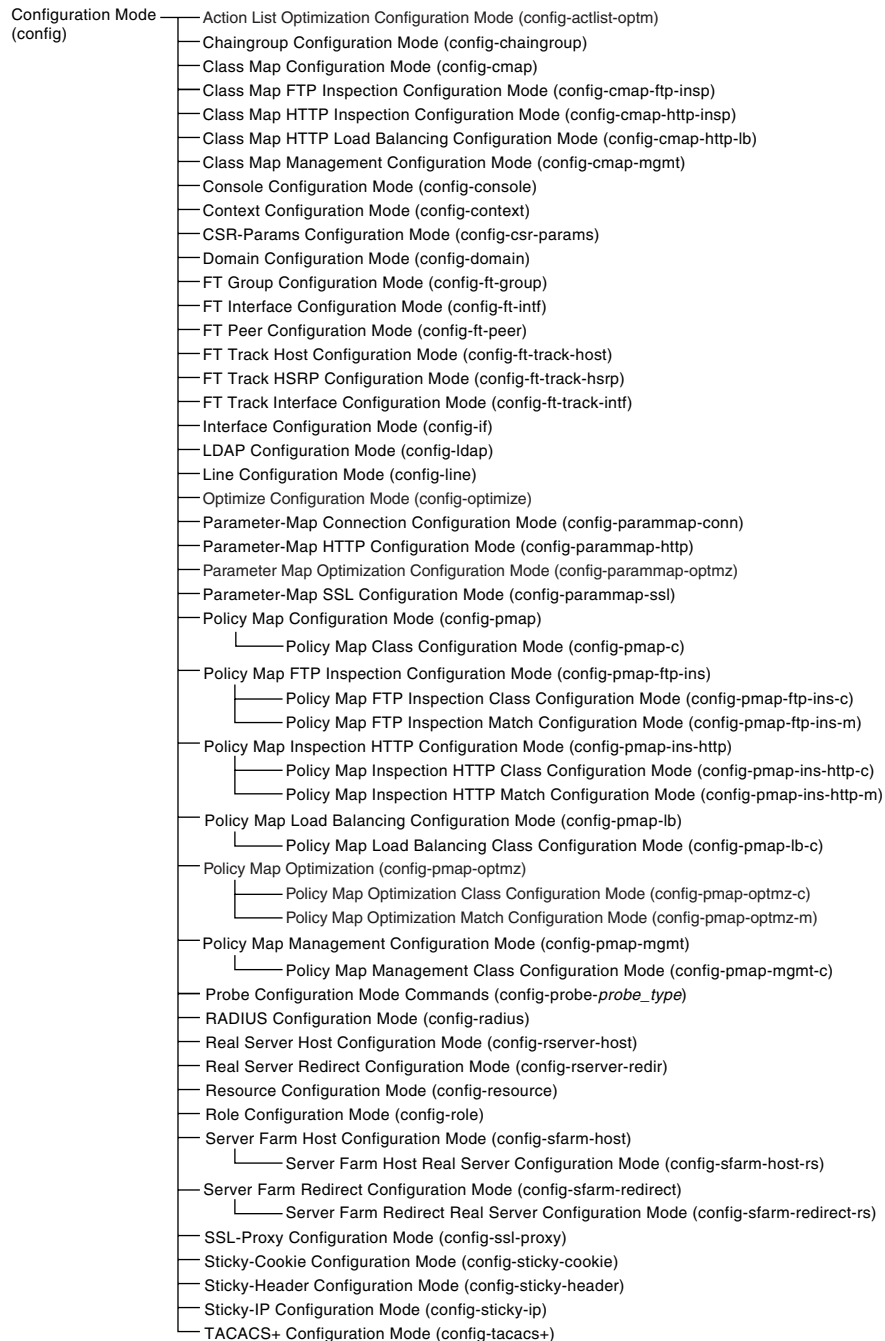
```
switch/Admin# configure  
switch/Admin (config)#
```

Configuration mode has a set of commands that allow you to configure the ACE and access its subordinate configuration modes. When you access any of the subordinate configuration modes, the ACE appends the mode name to the (config) prompt. For example, when you access real server host configuration mode from configuration mode, the prompt changes to (config-rserver-host).

To exit a configuration mode and access the previous mode, use the **exit** command. To exit any configuration mode and return to Exec mode, press **Ctrl-Z** or use the **end** command.

Figure 1-1 shows the modes that you can access from configuration mode and their associated prompts.

Figure 1-1 ACE Configuration Mode Hierarchy



240288

Using the CLI Commands

Table 1-1 lists CLI keyboard shortcuts to help you enter and edit command lines on the ACE. For further information on using the CLI commands, see the following sections:

- [Abbreviating Commands](#)
- [Editing the Command Line](#)
- [Defining Object Names](#)
- [Understanding the Effect of Contexts and Roles on Commands in Modes](#)
- [Using Exec Mode Commands in a Configuration Mode](#)
- [Understanding CLI Syntax Checking and Error Messages](#)

Table 1-1 CLI Command Keyboard Shortcuts

Action		Keyboard Shortcut
Cancel the current operation, delete the additional display of MORE output, or delete the current line.		Ctrl-c
Change:	The word at the cursor to lowercase.	Esc l
	The word at the cursor to uppercase.	Esc u
Delete:	A character to the left of the cursor.	Ctrl-h, Delete, or Backspace
	All characters from the cursor to the beginning of the line.	Ctrl-u
	All characters from the cursor to the end of the line.	Ctrl-k
	All characters from the cursor to the end of the word.	Esc d
	The word to left of the cursor.	Ctrl-w or Esc Backspace
Display the buffer's:	Next line.	Ctrl-n or Down-Arrow
	Previous line.	Ctrl-p or Up-Arrow
Display MORE output:	Exit from MORE output.	q, Q, or Ctrl-C
	Next additional screen. The default is one screen. To display more than one screen, enter a number before pressing the Spacebar key.	Spacebar
	Next line. The default is one line. To display more than one line, enter the number before pressing the Enter key.	Enter
Enter an Enter or Return key character.		Ctrl-m
Expand the command or abbreviation.		Ctrl-i or Tab

Table 1-1 CLI Command Keyboard Shortcuts (continued)

Action		Keyboard Shortcut
Move the cursor:	One character to the left (back).	Ctrl-b or Left Arrow
	One character to the right (forward).	Ctrl-f or Right Arrow
	One word to the left (back), to the beginning of the current or previous word.	Esc b
	One word to the right (forward), to the end of the current or next word.	Esc f
	To the beginning of the line.	Ctrl-a
	To the end of the line.	Ctrl-e
Redraw the screen at the prompt.		Ctrl-l or Ctrl-r
Return to the Exec mode from any configuration mode.		Ctrl-z
Return to the previous mode or exit from the CLI from Exec mode.		exit command
Transpose a character at the cursor with a character to the left of the cursor.		Ctrl-t

Abbreviating Commands

The ACE allows you to abbreviate most command keywords or options to its fewest unique characters. For example, instead of entering the full **write terminal** command, you can enter:

```
host/Admin# w t
Generating configuration....
```

Editing the Command Line

The ACE allows you to view all previously entered commands with the up arrow. Once you have examined a previously entered command, you can move forward in the list with the down arrow.

When you view a command that you wish to reuse, you can edit it or press the **Enter** key to execute it.

Defining Object Names

The following objects are user-configurable items:

- Access lists
- Class maps
- Defined interfaces
- Parameter maps
- Policy maps
- Health probes
- Real servers
- Server farms

- Scripts
- Sticky groups

The objects that you create are specific to the context where they are created. If the context is partitioned into multiple domains, you allocate objects within each domain.

The ACE supports case sensitivity when you configure an object name. If you create a context named C11 and another context c11, the ACE considers them as two different contexts. For example, enter:

```
host/Admin(config)# context C11
host/Admin(config-context)# exit
host/Admin(config)# context c11
host/Admin(config-context)# exit
```

When you perform a query for contexts, both C11 and c11 appear as separate contexts.

```
host/Admin(config)# context ?
<WORD> Enter the context name (Max Size - 64)
C11
c11
```

Understanding the Effect of Contexts and Roles on Commands in Modes

All commands are available to the Admin and the Admin context. However not all commands are available in user contexts. The ACE provides role-based access control (RBAC), which is a mechanism that determines the commands and resources available to each user. A role defines a set of permissions for accessing the objects and resources in a context and the actions that you can perform on them. The global or context administrator assigns roles to users based on their network function and the resources to which you want them to have access. For more information, see the *Cisco 4700 Series Application Control Engine Appliance Administration Guide*.

Using Exec Mode Commands in a Configuration Mode

When you are in a configuration mode, you may need to use a **show** command or any other command that is only available in Exec mode. To enter an Exec command in any configuration mode, use the **do** command. The syntax for this command is as follows:

```
do exec_command_string
```

The *exec_command_string* argument is the Exec mode command that you want to execute.

For example, to display the running configuration in configuration mode, enter:

```
host1/Admin(config)# do show running-config
```

Understanding CLI Syntax Checking and Error Messages

If you enter an invalid or incomplete command, the CLI responds with a pointer (^) and an error message. The following example shows the CLI response when you enter an invalid command:

```
host1/Admin# test
^
% invalid command detected at ^^ marker.
```

The following example shows the CLI response when you enter an incomplete command:

```
host1/Admin(config)# interface
                        ^
% incomplete command detected at '^' marker.
```

Getting CLI Help

The CLI provides several types of context-sensitive help, as described in the following sections:

- [Using the Question Mark \(?\)](#)
- [Using the Tab Key](#)

Using the Question Mark (?)

The question mark (?) character allows you to get the following type of help about a command at the command line:

Question Mark Usage	Command Help Type
? at command prompt	All commands for that mode
<i>command ?</i>	Any keywords, options, or object names for a command
<i>command keyword ?</i>	Any keywords, options, or object names for a command
<i>command-abbrev?</i>	All commands that begin with specific letters

Using the Tab Key

When you press the Tab key or Ctrl-I at the end of a unique command or option abbreviation, the CLI completes the command or options for you. For example:

```
host1/Admin# sh<Tab>
host1/Admin# show
```

Pressing the Tab key or Ctrl-I keys also completes an option up to the point where it is unique. If multiple commands have the same abbreviation that you entered, the CLI lists all of these commands.

Creating Configuration Files Using a Text Editor

This section describes how to format a text configuration file using a text editor that you can download to the ACE and contains the following topics:

- [How Commands Correspond with Lines in the Text File](#)
- [Subordinate Commands](#)
- [Automatic Text Entries](#)
- [Line Order](#)
- [Passwords](#)

To copy the file to the ACE, see *Cisco 4700 Series Application Control Engine Appliance Administration Guide*.

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide. In most cases, the command examples in this guide show how to enter the command preceded by the applicable CLI prompt. For example:

```
host1/Admin(config)# resource-class abc
```

In the text configuration file, you are not prompted to enter commands, so the prompt is omitted:

```
resource-class abc
```

Subordinate Commands

Subordinate commands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the subcommands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
resource-class abc
  limit-resource all minimum 0.00 maximum unlimited
  limit-resource rate syslog minimum 0.00 maximum unlimited
```

Automatic Text Entries

When you download a configuration file to the ACE, the ACE inserts some lines automatically. For example, the ACE inserts lines for default settings and for the date and time that the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as entries for access control lists (ACLs), are processed in the order that they appear, and the order can affect the function of the ACL. Other commands might also have order requirements. For example, you must enter the **interface vlan** command for an interface before you assign an IP address to it because many subsequent commands use the name of the interface. Also, subcommands must directly follow the main command.

Passwords

The user password is automatically encrypted before it is stored in the configuration. For example, the encrypted form of the letmein password might look like jMorNbK0514fadBh. You can copy the configuration passwords to another ACE in their encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the ACE does not automatically encrypt them when you copy the configuration to the ACE. The ACE encrypts them only when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.