



# Overview of Cisco UCS Manager CLI

---

This chapter includes the following sections:

- [Managed Objects, page 1](#)
- [Command Modes, page 1](#)
- [Object Commands, page 3](#)
- [Complete a Command, page 4](#)
- [Command History, page 4](#)
- [Committing, Discarding, and Viewing Pending Commands, page 4](#)
- [Online Help for the CLI, page 5](#)

## Managed Objects

Cisco UCS uses a managed object model, where managed objects are abstract representations of physical or logical entities that can be managed. For example, servers, chassis, I/O cards, and processors are physical entities represented as managed objects, and resource pools, user roles, service profiles, and policies are logical entities represented as managed objects.

Managed objects may have one or more associated properties that can be configured.

## Command Modes

The CLI is organized into a hierarchy of command modes, with the EXEC mode being the highest-level mode of the hierarchy. Higher-level modes branch into lower-level modes. You use **create**, **enter**, and **scope** commands to move from higher-level modes to modes in the next lower level, and you use the **exit** command to move up one level in the mode hierarchy.

**Note**

Most command modes are associated with managed objects, so you must create an object before you can access the mode associated with that object. You use **create** and **enter** commands to create managed objects for the modes being accessed. The **scope** commands do not create managed objects and can only access modes for which managed objects already exist.

Each mode contains a set of commands that can be entered in that mode. Most of the commands available in each mode pertain to the associated managed object. Depending on your assigned role and locale, you may have access to only a subset of the commands available in a mode; commands to which you do not have access are hidden.

The CLI prompt for each mode shows the full path down the mode hierarchy to the current mode. This helps you to determine where you are in the command mode hierarchy, and it can be an invaluable tool when you need to navigate through the hierarchy.

[Table 1: Main Command Modes and Prompts](#), [page 2](#) lists the main command modes, the commands used to access each mode, and the CLI prompt associated with each mode.

**Table 1: Main Command Modes and Prompts**

Mode Name	Commands Used to Access	Mode Prompt
EXEC	<b>top</b> command from any mode	#
chassis	<b>enter chassis</b> and <b>scope chassis</b> commands from EXEC mode	/chassis #
Ethernet server	<b>enter eth-server</b> and <b>scope eth-server</b> commands from EXEC mode	/eth-server #
Ethernet uplink	<b>enter eth-uplink</b> and <b>scope eth-uplink</b> commands from EXEC profile mode	/eth-uplink #
Fibre Channel uplink	<b>enter fc-uplink</b> and <b>scope fc-uplink</b> commands from EXEC profile mode	/fc-uplink #
firmware	<b>enter firmware</b> and <b>scope firmware</b> commands from EXEC profile mode	/firmware #
monitoring	<b>enter monitoring</b> and <b>scope monitoring</b> commands from EXEC profile mode	/monitoring #
organization	<b>create org</b> , <b>enter org</b> , and <b>scope org</b> commands from EXEC profile mode	/org #

Mode Name	Commands Used to Access	Mode Prompt
security	<b>enter security</b> and <b>scope security</b> commands from EXEC profile mode	/security #
fabric-interconnect	<b>enter fabric-interconnect</b> and <b>scope fabric-interconnect</b> commands from EXEC profile mode	/fabric-interconnect #
system	<b>enter system</b> and <b>scope system</b> commands from EXEC profile mode	/system #

## Object Commands

Four general commands are available for object management:

- **create** *object*
- **delete** *object*
- **enter** *object*
- **scope** *object*

You can use the **scope** command with any managed object, whether a permanent object or a user-instantiated object. The other commands allow you to create and manage user-instantiated objects. For every **create** *object* command, a corresponding **delete** *object* and **enter** *object* command exists.

In the management of user-instantiated objects, the behavior of these commands depends on whether the object exists, as described in the following tables:

**Table 2: Command behavior if the object does not exist**

Command	Behavior
<b>create</b> <i>object</i>	The object is created and its configuration mode, if applicable, is entered.
<b>delete</b> <i>object</i>	An error message is generated.
<b>enter</b> <i>object</i>	The object is created and its configuration mode, if applicable, is entered.
<b>scope</b> <i>object</i>	An error message is generated.

**Table 3: Command behavior if the object exists**

Command	Behavior
<b>create</b> <i>object</i>	An error message is generated.
<b>delete</b> <i>object</i>	The object is deleted.
<b>enter</b> <i>object</i>	The configuration mode, if applicable, of the object is entered.
<b>scope</b> <i>object</i>	The configuration mode of the object is entered.

## Complete a Command

You can use the Tab key in any mode to complete a command. Partially typing a command name and pressing Tab causes the command to be displayed in full, or to the point where another keyword must be chosen or an argument value must be entered.

## Command History

The CLI stores all previously used commands in the current session. You can step through the previously used commands by using the Up Arrow or Down Arrow keys. The Up Arrow key steps to the previous command in the history, and the Down Arrow key steps to the next command in the history. If you get to the end of the history, pressing the Down Arrow key does nothing.

All commands in the history can be entered again by simply stepping through the history to recall the desired command and pressing Enter. The command is entered as if you had manually typed it. You can also recall a command and change it before you enter it.

## Committing, Discarding, and Viewing Pending Commands

When you enter a configuration command in the CLI, the command is not applied until you enter the **commit-buffer** command. Until committed, a configuration command is pending and can be discarded by entering a **discard-buffer** command.

You can accumulate pending changes in multiple command modes and apply them together with a single **commit-buffer** command. You can view the pending commands by entering the **show configuration pending** command in any command mode.



### Note

Committing multiple commands together is not an atomic operation. If any command fails, the successful commands are applied despite the failure. Failed commands are reported in an error message.

While any commands are pending, an asterisk (\*) appears before the command prompt. The asterisk disappears when you enter the **commit-buffer** command, as shown in this example:

```
switch-1# scope chassis 1
switch-1 /chassis # enable locator-led
switch-1 /chassis* # show configuration pending
```

```
scope chassis 1
+   enable locator-led
exit
switch-1 /chassis* # commit-buffer
switch-1 /chassis #
```

## Online Help for the CLI

At any time, you can type the ? character to display the options available at the current state of the command syntax. If you have not typed anything at the prompt, typing ? lists all available commands for the mode you are in. If you have partially typed a command, typing ? lists all available keywords and arguments available at your current position in the command syntax.

