



## CLI Overview

---

- [Managed Objects, on page 1](#)
- [Command Modes, on page 1](#)
- [Object Commands, on page 3](#)
- [Complete a Command, on page 4](#)
- [Command History, on page 4](#)
- [Commit, Discard, and View Pending Commands, on page 4](#)
- [Inline Help for the CLI, on page 5](#)
- [CLI Session Limits, on page 5](#)

## Managed Objects

The Firepower eXtensible Operating System (FXOS) uses a managed object model, where managed objects are abstract representations of physical or logical entities that can be managed. For example, chassis, security modules, network modules, ports, and processors are physical entities represented as managed objects, and licenses, user roles, and platform 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 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 **up** command to move up one level in the mode hierarchy. You can also use the **top** command to move to the top 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.

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.

The following table 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	#
Adapter	<b>scope adapter</b> command from EXEC mode	/adapter #
Cabling	<b>scope cabling</b> command from EXEC mode	/cabling #
Chassis	<b>scope chassis</b> command from EXEC mode	/chassis #
Ethernet server domain	<b>scope eth-server</b> command from EXEC mode; this command and all subcommands are currently not supported	/eth-server #
Ethernet uplink	<b>scope eth-uplink</b> command from EXEC mode	/eth-uplink #
Fabric interconnect	<b>scope fabric-interconnect</b> command from EXEC mode	/fabric-interconnect #
Firmware	<b>scope firmware</b> command from EXEC mode	/firmware #
Host Ethernet interface	<b>scope host-eth-if</b> command from EXEC mode  <b>Note</b> This command and all subcommands are not supported at this level; the Host Ethernet interface commands are available in /adapter # mode.	/host-eth-if #
License	<b>scope license</b> command from EXEC mode	/license #
Monitoring	<b>scope monitoring</b> command from EXEC mode	/monitoring #
Organization	<b>scope org</b> command from EXEC mode	/org #
Security	<b>scope security</b> command from EXEC mode	/security #
Server	<b>scope server</b> command from EXEC mode	/server #

Mode Name	Commands Used to Access	Mode Prompt
Service profile	<b>scope service-profile</b> command from EXEC mode  <b>Note</b> Do not alter or configure service profiles; that is, do not use the <b>create, set, or delete</b> subcommand sets.	/service-profile #
SSA	<b>scope ssa</b> command from EXEC mode	/ssa #
System	<b>scope system</b> command from EXEC mode	/system #
Virtual HBA	<b>scope vhba</b> command from EXEC mode  <b>Note</b> This command and all subcommands are currently not supported.	/vhba #
Virtual NIC	<b>scope vnic</b> command from EXEC mode	/vnic #

## 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 you must enter another keyword or an argument value.

## Command History

The CLI stores all commands used 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 moves to the previous command in the history, and the down-arrow key moves to the next command in the history. When you get to the end of the history, pressing the down-arrow key does nothing.

You can enter any command in the history again by stepping through the history to recall that command and then pressing **Enter**. The command is entered as if you had manually typed it. You can also recall a command and change it before you press **Enter**.

## Commit, Discard, and View 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** All pending commands are checked for validity. However, if any queued command fails during commit, the remaining commands are applied; 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.

The following example shows how the prompts change during the command entry process:

```
Firepower# scope system
Firepower /system # scope services
Firepower /system/services # create ntp-server 192.168.200.101
Firepower /system/services* # show configuration pending
  scope services
+   create ntp-server 192.168.200.101
  exit
Firepower /system/services* # commit-buffer
Firepower /system/services #
```

## Inline Help for the CLI

At any time, you can enter the `?` character to display the options available at the current state of the command syntax.

If you have not entered anything at the prompt, entering `?` lists all available commands for the mode you are in. With a partially entered command, entering `?` lists all keywords and arguments available at your current position in the command syntax.

## CLI Session Limits

FXOS limits the number of CLI sessions that can be active at one time to 32 total sessions. This value is not configurable.

