



Using the CLI

- [About the ACI Switch CLI, on page 2](#)
- [iBash and VSH, on page 3](#)
- [Bash Conventions and Extensions, on page 4](#)

About the ACI Switch CLI

The ACI switch command-line interface (CLI) is iBash, which consists of the standard Bash command language interpreter shell plus a set of custom commands for ACI switches.



Note To reach the iBash shell, you must login as admin, and not as root.

iBash and VSH

Virtual Shell (VSH or vshell), the traditional Cisco NX-OS switch interface, is not a supported mode for ACI. Limited VSH functionality is present by typing `vsh` in the iBash shell, but its use is not intended for normal operation. This guide does not describe VSH.

In some situations, it could be necessary to use VSH to obtain process tags for use in an iBash command. For example, in NX-OS releases that support multiple OSPF instances, an iBash command such as `show ip ospf database` displays output only from the first registered OSPF process. To display output from other OSPF processes, you must include a process tag in the command: `show ip ospf <ptag> database`. You can view the running processes and obtain the process tag using a VSH command, as in the following example:

```
(none)# vsh -c "show cli internal ctags" | grep -i ospf
 4276 281    119 ospf      multipodIn 2450  ready    ospf-multipodInternal
 4278 281    119 ospf      default    2455  ready    ospf-default
 4277 282    128 ospfv3     default    2478  ready    ospfv3-default
```

To display the output of the OSPF default process in this example, use the iBash command `show ip ospf default database`.

Bash Conventions and Extensions

Bash (Bourne Again SHell) is a Unix shell or command-line interpreter supported by a variety of operating systems. You can use the Bash interface to directly access the ACI switches or you can develop Bash shell scripts to automate tasks. Bash provides a variety of command line and scripting features.

For more information about the Bash shell, see <http://www.gnu.org/software/bash/bash.html>.

Network Naming Conventions

Bash uses a forward slash (/) as a separator for interfaces, network addresses, and other settings.

For example, Ethernet interface number 46 on switch module 1 is represented as `Eth1/46`.

Command Completion

Bash provides tab completion for standard Linux commands and ACI switch-specific commands. When you press the **Tab** key at the end of a command or option abbreviation, Bash displays the command in full or the next available keyword or argument choice. Bash will also autocomplete unambiguous partial command arguments.



Note Bash will autocomplete partial commands without the **Tab** key only after you enter the first level command. For example, `show int br` will be autocompleted as `show interface brief`, but `sh int br` will not be autocompleted.

Command History

The CLI supports the Bash shell history functions. To display the command history, you can use the **Up Arrow** or **Down Arrow**, as well as the history command.

You can reenter a command in the history by stepping through the history to recall the desired command and pressing **Enter**. You can also recall a command and change it before you enter it.

In addition, you can directly search for a previous command by pressing **Ctrl-r** and then typing part of the desired command until the command is displayed.

Command Help

At any time, you can press the **Esc** key twice to display the options available at the current state of the command syntax. If you have not entered anything at the prompt, pressing the **Esc** key twice lists all available commands for the current command mode. If you have partially entered a command, pressing the **Esc** key twice lists all the keywords and arguments available at your current position in the command syntax.



Note The iBash CLI uses **Esc-ESC** instead of **?** to avoid a conflict with the use of **\$?** as a possible Bash prompt shortcut.
