CHAPTER 6

Understanding the CLI

This chapter provides information about the CLI in the following sections:

Information About the CLI Prompt

Once you have successfully accessed the system, the CLI prompt displays in the terminal window of your console port or remote workstation, as follows.

```
switch#
```

You can change this switch prompt to another name or leave it as it is.

Example:

```
switch# config t
switch(config)# switchname n1000v
n1000v(config)# exit
n1000v#
```

From the CLI prompt, you can do the following:

- Use CLI commands for configuring features.
- Access the command history.
- Use command parsing functions.

Command Modes

This section includes the following topics:

About Command Modes

Cisco Nexus 1000V CLI is divided into command modes which define the actions available to the user. Command modes are “nested” and are accessed in sequence. When you first log in, you are placed in CLI EXEC mode.

As you navigate from EXEC mode to global configuration mode, a larger set of commands are available to you. To transition to global configuration mode, enter the following command:

config t

The following table shows how command access builds from user EXEC to global configuration mode.

Command Mode	Prompt	Description
Exec	n1000v#	• Connect to remote devices. • Temporarily change terminal line settings. • Perform basic tests. • List system information (show).
Global Configuration	n1000v(config)#	• Configure features, such as the following: – port profile – VLANs – Interfaces • Includes access to EXEC commands. – Connect to remote devices. – Temporarily change terminal line settings. – Perform basic tests. – List system information (show).

All commands in EXEC command mode are accessible from the global configuration command mode. For example, the **show** commands are available from any command mode.

EXEC Command Mode

When you first log in, you are placed into EXEC mode. The commands available in EXEC mode include the **show** commands that display device status and configuration information, the **clear** commands, and other commands that perform actions that you do not save in the device configuration.

Global Configuration Command Mode

Global configuration mode provides access to the most broad range of commands, including those used to make configuration changes that are saved by the device, and can be stored and applied when the device is rebooted.

Commands entered in global configuration mode update the running configuration file as soon as they are entered, but must also be saved into the startup configuration file by using the following command:

copy running-config startup-config

In global configuration mode, you can access a number of protocol-specific, platform-specific, and feature-specific configuration modes.

Accessing Interface Configuration Command Mode

To access and list the interface configuration commands, follow these steps:

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number*
	
	Example:
	`n1000v(config)# interface ethernet 3/2`
	`n1000v(config-if)#`
		Enters interface configuration mode for the interface you want to configure.

For details about interface commands and configuration, see the document, *Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1)SV1(4)*.

Exiting a Configuration Mode

To exit from any Configuration mode, use any of the following commands:

Command	Purpose

exit
 	
Example:
`svs(config-if)# exit`
`svs(config)#`
	Exits from the current configuration command mode and return to the previous configuration command mode.

end
 	
Example:
`svs(config)# end`
`svs#`
	Exits from the configuration command mode and returns to EXEC mode.

Ctrl-z
 	
Example:
`svs(config)# ^z`
`svs#`
	Exits the current configuration command mode and returns to EXEC mode.
	
	Caution If you use Ctrl-Z at the end of a command line in which a valid command has been typed, the CLI adds the command to the running configuration file. We recommend that you exit a configuration mode using the **exit** or **end** command.

Command Mode Summary

Table 6-1 summarizes information about command modes.

Table 6-1 Command Mode Summary

Mode	Access Method	Prompt	Exit Method
EXEC	From the login prompt, enter your username and password.	`n1000v#`	To exit to the login prompt, use the **exit** command.
Global Configuration	From EXEC mode, enter the command, **configure terminal**.	`n1000v(config)#`	To exit to EXEC mode, use the **end** or **exit** command or press **Ctrl-Z**.
Port Profile Configuration	From Global Configuration mode, enter the command, **port-profile** *name*.	`n1000v(config-port-prof)#`	To exit to Port Profile Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
Interface Configuration	From Global Configuration mode, enter the **interface** command for a specific interface, for example, `interface veth 2`	`n1000v(config-if)#`	To exit to Interface Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
VLAN Configuration	Use a **vlan** command.	`n1000v(config-vlan)#`	To exit to VLAN Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
`Console Configuration`	From Global Configuration mode, use the **line console** command.	`n1000v(config-console)`	To exit to Console Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
Virtual Terminal Line Configuration	From Global Configuration mode, use the **line vty** command.	`n1000v(config-line)#`	To exit to Line Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
SVS Domain Configuration	From Global Configuration mode, use the **svs-domain** command.	`n1000v(config-svs-domain)#`	To exit to SVS Domain Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
Policy Map QoS Configuration	From Global Configuration mode, use the **policy-map** command.	`n1000v(config-pmap-qos)#`	To exit to Policy Map QoS Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.

Table 6-1 (continued)Command Mode Summary (continued)

Mode	Access Method	Prompt	Exit Method
Policy Map Class QoS Configuration	From Policy-Map QoS Configuration mode, use the class command.	`n1000v(config-pmap-c-qos)#`	To exit to Policy Map Class QoS Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
Class Map QoS Configuration	From Global Configuration mode, use the **class-map** command.	`n1000v(config-cmap-qos)#`	To exit to Class Map QoS Configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z.**

Saving CLI Configuration Changes

This section describes how changes you make using the CLI are saved and includes the following topics:

- Running Configuration, page 6-6
- Startup Configuration, page 6-6
- Copying the Running Configuration to the Startup Configuration, page 6-7

Running Configuration

The running configuration is the configuration that is currently running on the device. It includes configuration changes from commands entered since the last time the device was restarted. If the device is restarted, the running configuration is replaced with a copy of the startup configuration. Any changes that were made to the running configuration but were not copied to the startup configuration are discarded.

Startup Configuration

The startup configuration is the configuration that is saved and that will be used by the device when you restart it. When you make configuration changes to the device, they are automatically saved in the running configuration. If you want configuration changes saved permanently, you must copy them to the startup configuration so that they are preserved when the device is rebooted or restarted.

Copying the Running Configuration to the Startup Configuration

You can use this procedure to copy changes you have made to the running configuration into the startup configuration so that they are saved persistently through reboots and restarts.

	Command	Purpose
Step 1	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Special Characters

Table 6-2 lists the characters that have special meaning in Cisco Nexus 1000V text strings and should be used only in regular expressions or other special contexts.

Table 6-2 Special Characters

Character	Description
|	Vertical bar
< >	Less than or greater than

Keystroke Shortcuts

Table 6-3 lists command key combinations that can be used in both EXEC and configuration modes:

Table 6-3 Keystroke Shortcuts

Key(s)	Description
Ctrl-A	Moves the cursor to the beginning of the line
Ctrl-B	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Ctrl-C	Cancels the command and returns to the command prompt.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves the cursor to the end of the line.
Ctrl-F	Moves the cursor one character to the right.
Ctrl-G	Exits to the previous command mode without removing the command string.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L	Redisplays the current command line.
Ctrl-R	Redisplays the current command line.

Table 6-3 Keystroke Shortcuts (continued)

Key(s)	Description
Ctrl-T	Transposes the character to the left of the cursor with the character located to the right of the cursor.
Ctrl-U	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the word to the left of the cursor.
Ctrl-X, H	List history. When using this key combination, press and release the Ctrl and X keys together before pressing H.
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Ctrl-Z	Ends a configuration session, and returns you to EXEC mode. When used at the end of a command line in which a valid command has been typed, the resulting configuration is first added to the running configuration file.
↑	Displays the previous command in the command history.
↓	Displays the next command in the command history.
→ ←	Moves your cursor through the command history directionally to locate a command string.
?	Displays a list of available commands.

Tab	Completes the word for you after entering the first characters of the word, and then pressing the Tab key. All options that match are presented.
	Used to complete:
	• command names
	• scheme names in the file system
	• server names in the file system
	• file names in the file system
	Example
	`n1000v(config)# xm<Tab>`
	`n1000v(config)# xml <Tab>`
	`n1000v(config)# xml server`
	
	Example
	`n1000v(config)# c<Tab>`
	`callhome class-map clock cts`
	`cdp cli control-plane`
	
	`n1000v(config)# cl<Tab>`
	`class-map cli clock`
	`n1000v(config)# cla<Tab>`
	`n1000v(config)# class-map`

Table 6-3 Keystroke Shortcuts (continued)

Key(s)	Description
	Example n1000v# **cd bootflash:<Tab>** bootflash: bootflash://sup-1/ bootflash://sup-remote/ bootflash:/// bootflash://sup-2/ bootflash://sup-standby/ bootflash://module-5/ bootflash://sup-active/ bootflash://module-6/ bootflash://sup-local/ **Example** n1000v# **cd bootflash://mo<Tab>** bootflash://module-5/ bootflash://module-6/ n1000v# **cd bootflash://module-**

Abbreviating Commands

You can abbreviate commands and keywords by entering the first few characters of a command. The abbreviation must include sufficient characters to make it unique from other commands or keywords. If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Table 6-4 lists examples of command abbreviations.

Table 6-4 Examples of Command Abbreviations

Command	Abbreviation
configure terminal	**conf t**
copy running-config startup-config	**copy run start**
interface ethernet 1/2	**int e 1/2**
show running-config	**sho run**

Using the *No* Form of a Command

Almost every configuration command has a **no** form that can be used to disable a feature or function. For example, to remove a VLAN, use the **no vlan** command. To reenable it, use the plain **vlan** command form. The *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(4)* describes the **no** form of a command when available.

For example, if you use the **boot** command in global configuration mode, you can then use the **no boot** command undo the results:

```
n1000v(config)# boot system bootflash: svs1.bin
n1000v(config)# no boot system bootflash: svs1.bin
```

Using CLI Variables

The Cisco Nexus 1000V supports the definition and use of variables in CLI commands. You can use CLI variables as follows:

- Entered directly on the command line.
- Passed to the child script initiated using the **run-script** command. The variables defined in the parent shell are available for use in the child **run-script** command process (the "Running a Script" section on page 6-12).
- Passed as command line arguments to the **run-script** command (the "Running a Script" section on page 6-12).

CLI variables have the following characteristics:

- Cannot have nested references through another variable.
- Can persist across switch reloads.
- Can exist only for the current session

The Cisco Nexus 1000V software provides one predefined system variable, the TIMESTAMP variable.

User-Defined CLI Session Variables

You can define CLI session variables to persist only for the duration of your CLI session using the **cli var name** command in EXEC mode. CLI session variables are useful for scripts that you execute periodically.

The following example shows how to create a user-defined CLI session variable.

```
svs# cli var name testinterface ethernet 3/2
```

You can reference a variable using the syntax **$**(variable).

The following example shows how to reference a user-defined CLI session variable.

```
n1000v# show interface $(testinterface)
Ethernet3/2 is up
    Hardware is Ethernet, address is 0050.565a.2341 (bia 0050.565a.2341)
    MTU 1500 bytes, BW -332641784 Kbit, DLY 10 usec,
       reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA
    Port mode is trunk
    full-duplex, 1000 Mb/s
    Beacon is turned off
    Auto-Negotiation is turned on
    Input flow-control is off, output flow-control is off
    Rx
    222045 Input Packets 24263 Unicast Packets
    89347 Multicast Packets 108435 Broadcast Packets
    22529316 Bytes
    Tx
    33710 Output Packets 31393 Unicast Packets
    1898 Multicast Packets 419 Broadcast Packets 461 Flood Packets
    5221175 Bytes
    91323 Input Packet Drops 0 Output Packet Drops

n1000v#
```

Use the show cli variables command to display user-defined CLI session variables. The following example displays user-defined CLI session variables.

```
n1000v# show cli variables
VSH Variable List
-----------------
TIMESTAMP="2008-07-02-13.45.15"
testinterface="ethernet 3/1"
n1000v#
```

Use the **cli no var name** command to remove user-defined CLI session variables.

The following example removes a user-defined CLI session variable.

```
n1000v# cli no var name testinterface
```

User-Defined CLI Persistent Variables

You can define CLI variables that persist across CLI sessions and switch reloads using the **cli var name** command in configuration mode. These CLI persistent variables are defined in configuration mode and are saved in the running configuration file.

The following example shows how to create a user-defined CLI persistent variable.

```
n1000v# config t
n1000v(config)# cli var name mgmtport mgmt 0
n1000v(config)# exit
n1000v#
```

You can reference a variable using the syntax **$**(variable).

The following example shows how to reference a user-defined CLI persistent variable.

```
n1000v# show interface $(mgmtport)
mgmt0 is up
    Hardware is GigabitEthernet, address is 0000.0000.0000 (bia 0050.5681.5578)
    Internet Address is 10.78.1.63/24
    MTU 1500 bytes, BW 0 Kbit, DLY 0 usec,
       reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA
    full-duplex, 1000 Mb/s
    Beacon is turned off
    Auto-Negotiation is turned on
    Input flow-control is off, output flow-control is off
    321949 packets input, 67199373 bytes
    0 multicast frames, 0 compressed
    0 input errors, 0 frame, 0 overrun, 0 fifo
    30178 packets output, 7071526 bytes
    0 underrun, 0 output errors, 0 collisions
    0 fifo, 0 carrier errors

n1000v#
```

Use the **show cli variables** command to display user-defined CLI persistent variables.

The following example displays user-defined CLI persistent variables.

```
n1000v# show cli variables
VSH Variable List
-----------------
TIMESTAMP="2005-10-24-21.37.13"
mgmtport="mgmt 0"
```

Use the **no cli var name** command in configuration mode to remove user-defined CLI persistent variables.

The following example removes a user-defined CLI persistent variable.

```
n1000v# config t
n1000v(config)# cli no var name mgmtport
```

System-Defined Variables

Cisco Nexus 1000V supports one predefined variable: TIMESTAMP. This variable refers to the time of execution of the command in the format YYYY-MM-DD-HH.MM.SS.

Note The TIMESTAMP variable name is case sensitive. All letters must be uppercase.

The following example uses $(TIMESTAMP) when redirecting **show** command output to a file.

```
Example:
n1000v# show running-config > rcfg.$(TIMESTAMP)
n1000v# dir
       5718     Jul 02 14:09:58 2008  rcfg.2008-07-02-14.09.58

Usage for volatile://
       8192 bytes used
   20963328 bytes free
   20971520 bytes total
n1000v#
```

Working with Command Scripts

This section includes the following sections:

Running a Script

The **run-script** command executes the commands specified in a file. To use this command, be sure to create the file and specify commands in the required order.

Note You cannot create the script files at the switch prompt. You can create the script file on an external machine and copy it into the bootflash: directory. This section assumes that the script file resides in the bootflash: directory.

The syntax for this command is **run-script** *filename*.

This example displays the CLI commands specified in the file named *testfile* that resides in bootflash.

```
n1000v# show file bootflash:testfile
conf t
show interface mgmt 0
```

This file output is in response to the **run-script** command executing the contents in the testfile file:

```
pvk-s33# run-script bootflash:testfile
`conf t`
`show interface mgmt 0`
mgmt0 is up
Hardware: Ethernet, address: 0050.5682.4ace (bia 0050.5682.4ace)
Internet Address is 10.78.1.99/24
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA
full-duplex, 1000 Mb/s
Auto-Negotiation is turned on
25427 packets input, 2602757 bytes
0 multicast frames, 0 compressed
0 input errors, 0 frame, 0 overrun, 0 fifo
9077 packets output, 2433391 bytes
0 underrun, 0 output errors, 0 collisions
0 fifo, 0 carrier errors
...
```

Using CLI Variables in Scripts

You can use CLI variables defined by the **cli var** command or passed as arguments in the **run-script** command. For more information about the **cli var** command, see the "Using CLI Variables" section on page 6-10.

The following example shows how to use CLI session variables in a script file used by the **run-script** command.

```
n1000v# cli var name testinterface e 3/1

n1000v# show file bootflash:test1.vsh
show interface $(testvar)

n1000v# run-script bootflash:test1.vsh
`show interface $(testvar)`
Ethernet3/1 is down (Administratively down)
  Hardware is 10/100/1000 Ethernet, address is 0000.0000.0000 (bia 0019.076c.4da
c)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 0 bytes/sec, 0 packets/sec
  5 minute output rate 0 bytes/sec, 0 packets/sec
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
```

```
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
```

The following example shows how you can pass CLI session variable as arguments to a child **run-script** command process.

```
n1000v# show file bootflash:test1.vsh
show interface $(var1) $(var2)

n1000v# run bootflash:test2.vsh var1="e3/1" var2="brief"
`show interface $(var1) $(var2)`
--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth2/45       --     eth  routed down    Administratively down      auto(D) --
```

Delaying Command Action

The **sleep** command delays an action by a specified number of seconds, and is particularly useful within a script.

The syntax for this command is **sleep** *seconds*.

```
n1000v# sleep 30
```

You will the switch prompt return after 30 seconds.

Using Help

The CLI provides the following help features.

Feature	Description
?	You can type the question mark (?) to list the valid input options
^	The CLI prints the caret (^) symbol below a line of syntax to point to an input error in the command string keyword, or argument.
↑	You can use the up arrow to have the CLI display the previous command you entered so that you can correct an error.

The following example describes how to use syntax error isolation and context-sensitive help.

	Command	Purpose

Step 1	**show interface virtual ?**
	Example:
	`n1000v# show interface virtual ?`
	`  <CR>`
	`  >  Redirect it to a file`
	`  module  Limit display to interfaces on module`
	`  vm  Show interfaces owned by a Virtual Machine`
	`  vmk  Show interfaces owned by the Virtual Machine Kernel`
	`  vswif  Show interfaces owned by the Virtual Service Console`
	`  |  Pipe command output to filter`
	`n1000v# show interface virtual`
		Displays the optional parameters used with the **show interface virtual** command in EXEC mode.

Step 2	**show interface module ?**
	Example:
	`n1000v# show interface module ?`
	`  ^`
	`% invalid command detected at '^' marker.`
	`n1000v#`
		Displays an invalid command error message and points (^) to the syntax error.

Step 3	**Ctrl-P** or the Up Arrow
	Example:
	`n1000v# <Ctrl-P>`
	`n1000v# show interface virtual ?`
		Displays the previous command you entered so that you can correct the error.

Step 4	**show interface virtual module ?**
	Example:
	`n1000v# show interface virtual module ?`
	`  <1-256>  Enter module number`
	`n1000v# show interface virtual module`
		Displays the syntax for showing a virtual interface module.

Step 5	**show interface virtual module 3**
	Example:
	`n1000v# show interface virtual module 3`
	`---------------------------------------------`
	`Port  Adapter  Owner  Mod  Host`
	`---------------------------------------------`
	`n1000v#`
		Displays the virtual interface module 3.

	Command	Purpose

Step 6	**show module ?**
	Example:
	`n1000v# show module ?`
	`<CR>`
	`<1-66> Enter module number`
	`> Redirect it to a file`
	`internal Show line card manager related info`
	`uptime Show how long the module has been up and running`
	`vem Show Virtual Ethernet Module information`
	`| Pipe command output to filter`
		Displays the optional parameters for the show module command.

Step 7	**show module** **Example:** Example 6-1 on page 6-16.	Displays module information.

Example 6-1 Using Help

```
n1000v# show interface virtual ?
  <CR>
  >             Redirect it to a file
  module        Limit display to interfaces on module
  port-mapping  Show hypervisor port mapping
  vm            Show interfaces owned by a Virtual Machine
  vmk           Show interfaces owned by the Virtual Machine Kernel
  vswif         Show interfaces owned by the Virtual Service Console
  |             Pipe command output to filter
n1000v# show interface module ?
                               ^
% invalid command detected at '^' marker.
n1000v# <Ctrl-P>
n1000v# show interface virtual ?
n1000v# show interface virtual module ?
  <1-256>  Enter module number

n1000v# show interface virtual module ?
  <1-256>  Enter module number

n1000v# show interface virtual module 3

--------------------------------------------------------------------------------
Port        Adapter        Owner                    Mod Host
--------------------------------------------------------------------------------
n1000v# show module ?
  <CR>
  <1-32>    Enter module number
  >         Redirect it to a file
  internal  Show line card manager related info
  uptime    Show how long the module has been up and running
  |         Pipe command output to filter

n1000v# show module
show module
Mod  Ports  Module-Type                      Model              Status
---  -----  -------------------------------- ------------------ ------------
```

```
1    0      Virtual Supervisor Module        Nexus1000V         ha-standby
2    0      Virtual Supervisor Module        Nexus1000V         active *
3    248    Virtual Ethernet Module          NA                 ok
4    248    Virtual Ethernet Module          NA                 ok

Mod  Sw               Hw
---  ---------------  ------
1    4.0(4)SV1(0.33)  0.0
2    4.0(4)SV1(0.33)  0.0
3    4.0(4)SV1(0.33)  0.4
4    4.0(4)SV1(0.33)  0.4

Mod  MAC-Address(es)                          Serial-Num
---  --------------------------------------   ----------
1    00-19-07-6c-5a-a8 to 00-19-07-6c-62-a8   NA
2    00-19-07-6c-5a-a8 to 00-19-07-6c-62-a8   NA
3    02-00-0c-00-03-00 to 02-00-0c-00-03-80   NA
4    02-00-0c-00-04-00 to 02-00-0c-00-04-80   NA

Mod  Server-IP        Server-UUID                           Server-Name
---  ---------------  ------------------------------------  --------------------
1    10.78.1.99       NA                                    NA
2    10.78.1.99       NA                                    NA
3    10.78.1.92       8aca99de-16b7-300b-b572-730ea83c3de7  10.78.1.92
4    10.78.1.93       44454c4c-4800-104e-804d-b7c04f563153  10.78.1.93


* this terminal session
```

Displaying Available Features

To display a list of available features in Cisco Nexus 1000V and whether they are enabled on your device, use the **show feature** command from any command mode.

Example 6-2 Displaying Available Features

```
n1000v# show feature
Feature Name          Instance  State
--------------------  --------  --------
dhcp-snooping         1         enabled
http-server           1         enabled
ippool                1         enabled
lacp                  1         enabled
netflow               1         disabled
port-profile-roles    1         enabled
private-vlan          1         disabled
sshServer             1         enabled
tacacs                1         enabled
telnetServer          1         enabled
n1000v#
```