Scripting with Tcl

This chapter describes how to run tcl interactively and in scripts on a Cisco NX-OS device.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information" section or the "Feature History" table.

Guidelines and Limitations

Tcl has the following configuration guidelines and limitations:

Tclsh Command Help

Command help is not available for tcl commands. You can still access the help functions of Cisco NX-OS commands from within an interactive tcl shell.

This example shows the lack of tcl command help in an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# puts ?
          ^
% Invalid command at '^' marker.
switch-tcl# configure ?
  <CR>
  session   Configure the system in a session
  terminal  Configure the system from terminal input

switch-tcl#
```

Note In the above example, the Cisco NX-OS command help function is still available but the tcl **puts** command returns an error from the help function.

Tclsh Command History

You can use the arrow keys on your terminal to access commands you previously entered in the interactive tcl shell.

Note The **tclsh** command history is not saved when you exit the interactive tcl shell.

Tclsh Tab Completion

You can use tab completion for Cisco NX-OS commands when you are running an interactive tcl shell. Tab completion is not available for tcl commands.

Tclsh CLI Command

Although you can directly access Cisco NX-OS commands from within an interactive tcl shell, you can only execute Cisco NX-OS commands in a tcl script if they are prepended with the tcl **cli** command.

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli show module 1 | incl Mod
switch-tcl# cli "show module 1 | incl Mod"
switch-tcl# show module 1 | incl Mod
```

In a tcl script, you must prepend Cisco NX-OS commands with the tcl **cli** command as shown in this example:

```
set x 1
cli show module $x | incl Mod
cli "show module $x | incl Mod"
```

If you use the following commands in your script, the script will fail and the tcl shell will display an error:

```
show module $x | incl Mod
"show module $x | incl Mod"
```

Tclsh Command Separation

The semicolon (;) is the command separator in both Cisco NX-OS and tcl. To execute multiple Cisco NX-OS commands in a tcl command, you must enclose the Cisco NX-OS commands in quotes (""").

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli "configure terminal ; interface loopback 10 ; description loop10"
switch-tcl# cli configure terminal ; cli interface loopback 10 ; cli description loop10
switch-tcl# cli configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# cli interface loopback 10
switch(config-if-tcl)# cli description loop10
switch(config-if-tcl)#
```

In an interactive tcl shell, you can also execute Cisco NX-OS commands directly without prepending the tcl **cli** command:

```
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# description loop10
switch(config-if-tcl)#
```

Tcl Variables

You can use tcl variables as arguments to the Cisco NX-OS commands. You can also pass arguments into tcl scripts. Tcl variables are not persistent.

This example shows how to use a tcl variable as an argument to a Cisco NX-OS command:

```
switch# tclsh
switch-tcl# set x loop10
switch-tcl# cli "configure terminal ; interface loopback 10 ; description $x"
switch(config-if-tcl)#
```

Tclquit

The **tclquit** command exits the tcl shell regardless of which Cisco NX-OS command mode is currently active. You can also press **Ctrl-C** to exit the tcl shell. The **exit** and **end** commands change Cisco NX-OS command modes. The **exit** command will terminate the tcl shell only from the EXEC command mode.

Tclsh Security

The tcl shell is executed in a sandbox to prevent unauthorized access to certain parts of the Cisco NX-OS system. The system monitors CPU, memory, and file system resources being used by the tcl shell to detect events such as infinite loops, excessive memory utilization, and so on.

You configure the intial tcl environment with the **scripting tcl init** *init-file* command.

You can define the looping limits for the tcl environment with the **scripting tcl recursion-limit** *iterations* command. The default recursion limit is 1000 interations.

Information about Tcl

Tool Command Language (Tcl) is a scripting language created by John Ousterhout at the University of California, Berkeley. Tcl 8.5 was added to Cisco NX-OS Release 5.1(1) to provide scripting abilities. With tcl, you gain more flexibility in your use of the CLI commands on the device. You can use tcl to extract certain

values in the output of a **show** command, perform switch configurations, run Cisco NX-OS commands in a loop, or define EEM policies in a script.

This section describes how to run tcl scripts or run tcl interactively on Cisco NX-OS devices.

Running the tclsh Command

You can run tcl commands from either a script or on the command line using the **tclsh** command.

✎

Note   You cannot create a tcl script file at the CLI prompt. You can create the script file on a remote device and copy it to the bootflash: directory on the Cisco NX-OS device.

Procedure

	Command or Action	**Purpose**

Step 1	**tclsh** [**bootflash:***filename* [*argument* ...]]
	
	Example:
	 ```switch# tclsh ?```
	  `<CR>`
	  `bootflash:  The file to run`
		Starts a tcl shell.
		
		 If you run the **tclsh** command with no arguments, the shell runs interactively, reading tcl commands from standard input and printing command results and error messages to the standard output. You exit from the interactive tcl shell by entering **tclquit** or pressing **Ctrl-C**.
		
		 If you enter the **tclsh** command with arguments, the first argument is the name of a script file that contains tcl commands and any additional arguments are made available to the script as variables.

Example

This example shows an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# cli show module $x | incl Mod
Mod  Ports  Module-Type                  Model              Status
1    32     1/10 Gbps Ethernet Module    N7K-F132XP-15      ok
Mod  Sw          Hw
Mod  MAC-Address(es)                  Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch-tcl# exit
switch#
```

This example shows how to run a tcl script:

```
switch# show file bootflash:showmodule.tcl
set x 1
while {$x < 19} {
cli show module $x | incl Mod
```

```
set x [expr {$x + 1}]
}

switch# tclsh bootflash:showmodule.tcl
Mod  Ports  Module-Type                    Model            Status
1    32     1/10 Gbps Ethernet Module      N7K-F132XP-15    ok
Mod  Sw          Hw
Mod  MAC-Address(es)                        Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch#
```

Navigating Cisco NX-OS Modes from the tclsh Command

You can change modes in Cisco NX-OS while you are running an interactive tcl shell.

Procedure

	Command or Action	**Purpose**

Step 1	**tclsh**
	
	Example:
	`switch# tclsh`
	`switch-tcl#`
		Starts an interactive tcl shell.

Step 2	**configure terminal**
	
	Example:
	`switch-tcl# configure terminal`
	`switch(config-tcl)#`
		Runs a Cisco NX-OS command in the tcl shell, changing modes.
		
		Note The tcl prompt changes to indicate the Cisco NX-OS command mode.

Step 3	**tclquit**
	
	Example:
	`switch-tcl# tclquit`
	`switch#`
		Terminates the tcl shell and returns to the starting mode.

Example

This example shows how to change Cisco NX-OS modes from an interactive tcl shell:

```
switch# tclsh
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# ?
  description  Enter description of maximum 80 characters
  inherit      Inherit a port-profile
  ip           Configure IP features
  ipv6         Configure IPv6 features
  logging      Configure logging for interface
  no           Negate a command or set its defaults
```

```
       rate-limit   Set packet per second rate limit
       shutdown     Enable/disable an interface
       this         Shows info about current object (mode's instance)
       vrf          Configure VRF parameters
       end          Go to exec mode
       exit         Exit from command interpreter
       pop          Pop mode from stack or restore from name
       push         Push current mode to stack or save it under name
       where        Shows the cli context you are in

switch(config-if-tcl)# description loop10
switch(config-if-tcl)# tclquit
Exiting Tcl
switch#
```

Tcl References

The following titles are provided for your reference:

- Mark Harrison (ed), *Tcl/Tk Tools*, O'Reilly Media, ISBN 1-56592-218-2, 1997

- Mark Harrison and Michael McLennan, *Effective Tcl/Tk Programming*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63474-0, 1998

- John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63337-X, 1994.

- Brent B. Welch, *Practical Programming in Tcl and Tk*, Prentice Hall, Upper Saddle River, NJ, USA, ISBN 0-13-038560-3, 2003.

- J Adrian Zimmer, *Tcl/Tk for Programmers*, IEEE Computer Society, distributed by John Wiley and Sons, ISBN 0-8186-8515-8, 1998.