



# Creating a TCL script to augment the functionality of the IOS CLI

---

By Boštjan Šuštar, CCIE #2387

21. 10. 2014

This document describes how to create TCL scripts to capture and execute Cisco IOS CLI commands. The example provided show how it is possible to add functionality to the CLI by allowing a simple "show" command within subconfiguration modes to display the portion of the configuration pertaining to the active subconfiguration mode.

## Managing Scripts

The major steps in installing TCL scripts:

1. Create a directory to store TCL scripts
2. Upload scripts to that directory
3. Register scripts

### 1. Create a directory to store TCL scripts

Create a directory on the router's storage (e.g. flash; replace "flash" with whatever your router uses).

```
MyRouter# mkdir flash:tcl  
MyRouter# conf t  
MyRouter(config)# event manager directory user policy "flash:/tcl"
```

## 2. Upload scripts

Use SCP or another supported method to upload the scripts to the router.

## 3. Register scripts

The scripts below are described in the next section.

```
MyRouter(config)# event manager policy cliparser.tcl type user  
MyRouter(config)# event manager policy cliaddon.tcl type user
```

# Sample Script - TCL script augmenting the configuration mode CLI

## Description

Issue a show command within a subconfiguration mode to display the current configuration pertaining to that subconfiguration mode.

## Solution

The solution is divided into the following steps:

1. The first script captures changes in the configuration level for a set of typical subconfiguration modes (e.g. interface, router, vrf, class map, policy map), issues the appropriate show running command (e.g. "show running-config | section router bgp") and stores the result.
2. The second step is to configure aliases to map the "show" (or whatever other alias you prefer) to start the second script for all the required configuration modes (same as those in the first script).
3. The second script displays the stored contents of the first script when the alias is executed in one of the configured subconfiguration modes.
4. The last step is to register the scripts.

**Script 1** (*cliparser.tcl*): Capture configuration mode commands that switch to a subconfiguration mode and store the configuration of that object to a file.

```

::cisco::eem::event_register_cli sync yes occurs 1 pattern "^interface
(.*?)$|^router (.*?)$|^ip vrf (.*?)$|^class-map (.*?)$|^policy-map
(.*?)$|^archive$|^redundancy$|^vlan$|^ip access-list (.*?)$|^line
(.*?)$|^route-map (.*?)$"

namespace import ::cisco::eem::*

namespace import ::cisco::lib::*

array set evtData [event_reqinfo]

set _cli_msg $evtData(msg)

#

# INTERFACE CONFIGURATION COMMAND

#

if { [ regexp -nocase {^interface (.*?)$} $_cli_msg ignore intf] } {

    appl_reqinfo key "cmd"

    appl_setinfo key "cmd" data "show running interface $intf"

    exit 1

}

#

# ROUTER CONFIGURATION COMMAND

#

if { [ regexp -nocase {^router (.*?)$} $_cli_msg ignore rtr] } {

    appl_reqinfo key "cmd"

    appl_setinfo key "cmd" data "show running | section router $rtr"

    exit 1

}

#

# VRF CONFIGURATION COMMAND

#

```

```
if { [ regexp -nocase {^ip vrf (.*)$} $_cli_msg ignore vrf] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section vrf $vrf"
    exit 1
}

#
# CLASS MAP CONFIGURATION COMMAND
#
if { [ regexp -nocase {^class-map (.*)$} $_cli_msg ignore cm] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section ^class-map $cm"
    exit 1
}

#
# POLICY MAP CONFIGURATION COMMAND
#
if { [ regexp -nocase {^policy-map (.*)$} $_cli_msg ignore pm] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section policy-map $pm"
    exit 1
}

#
# ARCHIVE CONFIGURATION COMMAND
#
if { [ regexp -nocase {^archive$} $_cli_msg ignore] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section ^archive\$"
    exit 1
}
```

```
#
# ACL CONFIGURATION COMMAND
#
if { [ regexp -nocase {^ip access-list (.*)$} $_cli_msg ignore acl] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section ip access-list $acl"
    exit 1
}

#
# LINE CONFIGURATION COMMAND
#
if { [ regexp -nocase {^line (.*)$} $_cli_msg ignore line] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section line $line"
    exit 1
}

#
# ROUTE MAP CONFIGURATION COMMAND
#
if { [ regexp -nocase {^route-map (.*)$} $_cli_msg ignore rm] } {
    appl_reqinfo key "cmd"
    appl_setinfo key "cmd" data "show running | section route-map $rm"
    exit 1
}

...

```

**Aliases:** Capture the show command and display the configuration of the subconfiguration mode.

```
alias ipenacl show do event manager run cliaddon.tcl
alias ipsnacl show do event manager run cliaddon.tcl
alias vrf show event manager run cliaddon.tcl
alias qospolycymap show do event manager run cliaddon.tcl
alias qosclassmap show do event manager run cliaddon.tcl
alias route-map show do event manager run cliaddon.tcl
alias router show do event manager run cliaddon.tcl
alias archive show do event manager run cliaddon.tcl
alias line show do event manager run cliaddon.tcl
alias interface show do event manager run cliaddon.tcl
...
```

**Script 2** (*cliaddon.tcl*): Capture the show command and display the configuration of the subconfiguration mode.

```
::cisco::eem::event_register_none sync yes
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

set cmda [appl_reqinfo key "cmd"]
if { [ lindex $cmda 0 ] == "data" } {
    set cmd [ lindex $cmda 1 ]
    appl_setinfo key "cmd" data $cmd
}
if [catch {cli_open} result] {
    puts stderr "%ACME-1-EXEC: CLI OPEN ($result)"
    exit 0
} else {
    array set cfd $result
    set fd $result
}
```

```
}  
  
catch {cli_exec $cfd(fd) "en"} result  
  
catch {cli_exec $cfd(fd) $cmd} result  
  
puts $result  
  
if [catch {cli_close $cfd(fd) $cfd(tty_id)} result] {  
    puts stderr "%ACME-1-EXEC: CLI CLOSE ($result)"  
    exit 0  
}  
  
exit 0
```

**Register scripts:** Configure EEM and register the scripts.

```
event manager directory user policy "flash:/tcl"  
  
event manager policy cliparser.tcl type user  
  
event manager policy cliaddon.tcl type user
```

**Note:** This is far from an optimal way of accomplishing the requirements but it is a good starting point for learning how to handle the CLI using TCL. The main issues with this implementation are:

- The "show" alias will not show the subconfiguration items created since the time the subconfiguration mode was entered.
- The command to enter the subconfiguration mode must sometimes use complete commands in order to work properly (e.g. using "class-m MyMap" or "class-map MyMap" will not work; you have to use the complete command "class-map match-any MyMap").

**Usage:** The following capture shows how to make use of the new functionality.

```
acme-rtr# conf t  
  
Enter configuration commands, one per line. End with CNTL/Z.  
  
acme-rtr(config)# interface fast4  
  
acme-rtr(config-if)# show  
  
Building configuration...
```

```
Current configuration : 367 bytes
!
interface FastEthernet4
  description The Net
  ip address 23.75.345.200 255.255.255.0
  ip access-group FW in
  no ip proxy-arp
  ip nat outside
  ip inspect FW in
  ip inspect FW out
  snmp ifindex persist
  no cdp enable
  crypto map CM
end

acme-rtr#

acme-rtr(config-if)# exit
acme-rtr(config)# line vty 0 4
acme-rtr(config-line)# show

line vty 0 4
  exec-timeout 20 0
  privilege level 15
  logging synchronous
  login authentication VTY
  length 30
  escape-character 3

acme-rtr#

acme-rtr(config-line)#
```



```
acme-rtr(config-line)# exit
acme-rtr(config)#
acme-rtr(config)# ip access-list ex MGMT
acme-rtr(config-ext-nacl)# show
ip access-list extended MGMT
  permit tcp any any eq 22
  permit udp any any eq snmp
acme-rtr#

acme-rtr(config-ext-nacl)# exit
acme-rtr(config)#
```

## NIL – More Than Just a Training Company

NIL Learning delivers the leading-edge Cisco training to IT professionals and companies around the globe. Through field-proven experts — each both active engineer and instructor — NIL Learning enhances the standard learning curriculum with real-life experience and helps clients to maximize their training investment.

NIL Learning is part of NIL, a leading global IT solutions provider. Since 1992, NIL has been at the forefront of advanced contributors to strategic partner Cisco's technologies, learning curriculum and value-added solutions deployed to clients around the globe. Today, NIL has earned the highest certifications offered by Cisco, VMware, EMC, HP, IBM, Microsoft, F5, Jive, MobileIron, RSA, VCE and others. Their portfolio of solutions consists of managed services, professional services and learning services.

NIL is headquartered in Slovenia, with regional offices in Croatia, Serbia, Saudi Arabia, the U.S., Turkey, South Africa, Morocco, Nigeria, Kenya and Botswana.

## Why Learn at NIL LEARNING?

- All NIL LEARNING instructors are **field-proven experts** - each both active engineer, content developer and instructor.
- **75% of NIL LEARNING engineers hold CCSI certifications**, and **20 have already achieved the respected CCIE rank**.
- NIL LEARNING **enhances the standard learning curriculum with real-life experience** and helps clients to maximize their training investment.
- NIL has been a Cisco training partner since 1993; today NIL holds **Cisco Learning Partner Specialized** status and **Cisco Business Learning Partner** status.
- NIL was awarded the **Cisco Most Business Relevant Learning Partner in MEA** in 2010 and the **Most Innovative Learning Partner** in MEA.

- NIL received the **Innovation Award** for its Technology Led Training and its extensive contribution to Cisco learning solutions at the Cisco EMEAR Learning Partner Summit in 2012.
- NIL received the **Innovation Award** for its Technology Led Training and Advanced Engineer Program at the Cisco Global Learning Partner Summit in 2013.
- NIL won the **Cisco Global Learning Partner of the Year** award at the Cisco Partner Summit in 2014.
- NIL Learning runs a centralized training schedule across the whole EMEA region.

## More Info

### Slovenia

T: +386 1 4746 500

E: [sales-support@nil.com](mailto:sales-support@nil.com)

### Saudi Arabia

T: +966 1 465 4641

E: [info.nilme@nil.com](mailto:info.nilme@nil.com)

### Botswana

T: +267 318 1684

E: [training@it-ig.bw](mailto:training@it-ig.bw)

### Serbia

T: +381 11 2282 818

E: [info-nilserbia@nil.co.rs](mailto:info-nilserbia@nil.co.rs)

### Croatia

T: +385 (0)51 583 255

E: [info-nilcroatia@nil.com](mailto:info-nilcroatia@nil.com)

### South Africa

T: +27 (0)11 575 4637

E: [mea\\_sales@nil.com](mailto:mea_sales@nil.com)

### Kenya

T: +27 (0)11 575 4637

E: [mea\\_sales@nil.com](mailto:mea_sales@nil.com)

### Turkey

T: +902 123 81 8639

E: [info-nilturkey@nil.com](mailto:info-nilturkey@nil.com)

### Morocco

T: +212(0) 660 808 394

E: [info-nilmorocco@nil.com](mailto:info-nilmorocco@nil.com)

### USA

T: +1 612 886 3900

E: [info-nilusa@nil.com](mailto:info-nilusa@nil.com)

### Nigeria

T: +27 (0)11 575 4637

E: [mea\\_sales@nil.com](mailto:mea_sales@nil.com)

[learning.nil.com](http://learning.nil.com)