CLI Configuration Commands

Use the CLI configuration commands to modify and then activate a device's configuration parameters.

To enter configuration mode, type the **config** command in operational mode. All changes to the device's configuration are made to a copy of the active configuration, called a candidate configuration. These changes do not take effect until you issue a successful **commit** or **commit confirm** command.

CLI Operational Commands

Use the CLI operational commands to view system status, monitor and troubleshoot a Cisco vEdge device and network connectivity, initiate configuration mode, and control the CLI environment. When you first enter the CLI, you are in operational mode.

CLI Overview

The CLI on the Cisco vEdge devices is one of the ways you can configure and monitor these devices. The CLI provides various commands for configuring and monitoring the software, hardware, and network connectivity of the vSmart controllers and the vEdge routers. The CLI provides the following features:

- Displaying help about CLI commands
- Completing partial commands
- Editing the command line with keyboard sequences
- Configuring CLI session settings
- Filtering command output
- Adding comments to device configurations
- Activating and deactivating parts of a configuration
- Displaying CLI messages

The Cisco SD-WAN CLI design is based on the YANG data modeling language, defined in RFC 6020.

CLI Modes

The CLI has two modes:

- Operational mode, for monitoring the state of the Cisco vEdge device. When you log in to the CLI, you are in operational mode. In this mode, you view device status, monitor and troubleshoot the device and network connectivity, enter into configuration mode, and control the CLI session parameters.
- Configuration mode, for changing the operational parameters of the Cisco vEdge device. You enter configuration mode by issuing the configure command in operational mode. This mode has a number of submodes for manipulating different parts of the configuration. For example, the mode interface-eth1 allows you to configure parameters for Ethernet interface 1. All changes to the device's configuration are done to a copy of the active configuration, called a candidate configuration. Configuration changes take effect only when you enter a commit or commit confirmed command and that command is successful.

Start the CLI

Before you begin, make sure the vSmart controller and the vEdge router hardware is set up and the Cisco SD-WAN software is installed. You must have a direct console connection to the device or network using SSH. If your device is not set up, follow the installation instructions provided to you with the vSmart controller or the vEdge router before proceeding.

The login prompt for a Cisco vEdge device shows the software version and then prompts for a username and password.

When you log into a vSmart controller or a vEdge router, you are prompted to enter your user name and password. Once you enter your password, you are automatically placed at the CLI prompt.

For security reasons, each time you log out of the device, the CLI session ends and you are required to log in again to access the CLI.

CLI Prompts

The prompt indicates the mode the CLI is in:

- *host-name#*: The host name followed by a hash mark indicates that the CLI is in operational mode. An operational mode prompt is similar to vsmart#.
- *host-name*(config)#: When the CLI is in configuration mode, the string config is added to the prompt. For example, a configuration mode prompt is similar to vsmart(config)#. If you are configuring a lower hierarchy in the commands, the prompt also indicates that level. For example, if you are configuring Ethernet interface 1 for a VPN, in the hierarchy vpn > interface, the configuration mode prompt is vsmart(config-interface-eth1)#. The CLI prompt shows only the parent hierarchy, not the full path to the command, so that the CLI prompt never gets too long.

To change the operational mode prompt, use the **prompt1** operational command:

```
vsmart# prompt1 eve@vsmart#
eve@vsmart#
```

To change the configuration mode prompt, use the **prompt2** operational command:

```
vsmart# prompt2 eve@vsmart(config)#
eve@vsmart(config)#
```

Configure CLI Session Settings

The following are the default CLI session settings for a Linux terminal:

```
vsmart# show cli
autowizard            false
complete-on-space     false
history               100
idle-timeout          1800
ignore-leading-space  true
output-file           terminal
paginate              true
prompt1               \h\M#
prompt2               \h(\m)#
screen-length         30
screen-width          80
service prompt config true
show-defaults         false
terminal              xterm-256color
timestamp             disable
```

To change the session values, use the command names listed in the output above. For more information on the commands, see Operational Commands .

Command Hierarchies

CLI commands are organized in a hierarchy that groups commands that perform related or similar functions. For example, in operational mode, commands that display information about OMP are collected under the **show omp** command hierarchy. In configuration mode, commands that configure OMP properties are collected under the **omp** command hierarchy.

Display Help about CLI Commands

To list the available CLI commands, along with a short description of the command, type a **?** (question mark).

If you type **?** at the prompt, the CLI displays a list of available commands. In operational mode, you see:

```
vsmart# ?
Possible completions:
  autowizard          Automatically query for mandatory elements

  clear               Clear parameter

  clock               System clock

  commit              Confirm a pending commit

  complete-on-space   Enable/disable completion on space

  config              Manipulate software configuration information

  debug               Debugging commands

  exit                Exit the management session

  file                Perform file operations

  help                Provide help information

  history             Configure history size

  idle-timeout        Configure idle timeout

  job                 Job operations

  leaf-prompting      Automatically query for leaf values
```

```
  logout              Logout a user
  monitor             Monitor a file
  no                  Negate a command or set its defaults
  nslookup            Look up a DNS name
  paginate            Paginate output from CLI commands
  ping                Ping a host
  poweroff            Shut down the system
  prompt1             Set operational mode prompt
  prompt2             Set configure mode prompt
  quit                Exit the management session
  reboot              Reboot the system
  request             Perform an action
  screen-length       Configure screen length
  screen-width        Set CLI screen width
  show                Show information about the system
  tcpdump             Perform tcpdump on a network interface
  timestamp           Enable/disable the display of timestamp
  tools               Tools commands
  traceroute          Trace connectivity to a host
  vdig                Asynchronous FQDN resolution
  vping               Send L2, L3, L7 probes to remote host
  vshell              System shell
```

If you type tools and **?** at the prompt, the CLI displays a list of available commands for tools. In operational mode, you see:

```
vm9# tools ?
Possible completions:
  consent-token   Access restricted functionality using Consent Token
  core-state      Show Core state
  cpu-util        Show CPU Utilization
  flood-ping      Flood-ping a host
  ike-debug       IKE debug tools
  internal        (TESTBED) Internal commands
  ip-route        Display route table
```

```
  iperf           Network bandwidth measurement tool

  netstat         Display network status

  nping           Network packet generation tool

  ss              Display network statistics

  stun-client     STUN client protocol tool

  support         Support commands

  vtysh           Integrated shell for Quagga routing software suite
```

Note To access **vtysh** commands, see *Quagga docs* on the Quagga Routing website.

If you type **?** at the prompt after entering configuration mode, you see:

```
vsmart(config)# ?
Possible completions:
  apply-policy   Apply network policy
  banner         Set banners
  omp            OMP information
  policy         Configure policy
  security       Configure security
  snmp           Configure SNMP
  system         Configure System
  vpn            VPN Instance
  ---
  abort          Abort configuration session
  clear          Remove all configuration changes
  commit         Commit current set of changes
  describe       Display transparent command information
  do             Run an operational-mode command
  end            Terminate configuration session
  exit           Exit from current mode
  help           Provide help information
  load           Load configuration from an ASCII file
  no             Negate a command or set its defaults
  pwd            Display current mode path
  revert         Copy configuration from running
  rollback       Roll back database to last committed version
  save           Save configuration to an ASCII file
  show           Show a parameter
  top            Exit to top level and optionally run command
  validate       Validate current configuration
```

If you type **?** after a command name, the CLI shows all possible completions for that command. For example:

```
vsmart# show interface vpn 0 ?
Possible completions:
  eth0  eth1  |  <>
```

If you type **help** before a command name, it will you give you more information about the command. For example:

```
vsmart# help show cli
Help for command: show cli
    Display cli settings
```

The **show parser dump** command also displays information about available commands and their syntax.

Enter User-Defined Strings

For many configuration commands, you define a string that identifies an instance of a configurable object. For example, when you create user accounts, you configure a user-defined string for the username:

```
vEdge(config-system)# aaa user eve
```

In this command, the strings "aaa" and "user" are Cisco SD-WAN software keywords, and the string "eve" is a user-defined string.

User-defined strings can include all uppercase and lowercase letters, all digits, spaces, and all special characters except for angle brackets (< and >).

To include a space or an exclamation point (!) in a user-defined string, either type a backslash (\) before the space or enclose the entire string in quotation marks (" "). For example:

```
vEdge(config)# banner login "Remember to log out when you are done!"
vEdge(config-banner)# show full-configuration
banner
 login "Remember to log out when you are done!"
!
vEdge(config-banner)#

vEdge(config-system)# organization-name My\ Company
vEdge(config-system)# show configuration
system
 organization-name "My Company"
!
vEdge(config-system)#
```

Complete Partial Commands and Strings

The CLI provides command completion. It recognizes commands and options based on the first few letters you type so that you do not always have to remember or type the full command or option name.

To display a list of all possible command or option completions, type the partial command followed immediately by a question mark. For example:

```
vsmart@# s?
Possible completions:
  screen-length        Configure screen length
  screen-width         Set CLI screen width
  show                 Show information about the system
```

To complete a command or option that you have partially typed, press the tab key after you have typed a partially completed command name. If the partially typed letters begin a string that uniquely identifies a command, the complete command name is displayed. Otherwise, a list of possible completions is displayed.

Command completion also works with other strings, such as filenames, directory names, interface names, and usernames.

To enable command completion when you press the space bar, enable it for the duration of the terminal session:

```
vEdge# complete-on-space true
```

When this is enabled, you can press the tab key or the space bar to complete a partially typed command name or variable string.

Command completion is disabled within quoted strings. So if an argument contains spaces and you quote them with a backslash (for example, **prefix-list my\ list**) or with quotation marks (for example, **prefix-list "my list"**), you cannot use command completion. Space completion does not work with filenames.

Edit the Command Line with Keyboard Sequences

You can use keyboard sequences in the CLI to move around and edit text on the command line itself. You can also use keyboard sequences to scroll through a list of recently executed commands. The following table lists some of the CLI keyboard sequences.

Table 1:

Category	Action	Keyboard Sequence
Move the cursor	Move the cursor back one character.	Ctrl-B or Left Arrow
	Move the cursor back one word.	Esc-B or Alt-B
	Move the cursor forward one character.	Ctrl-F or Right Arrow
	Move the cursor forward one word.	Esc-F or Alt-F
	Move the cursor to the beginning of the command line.	Ctrl-A or Home
	Move the cursor to the end of the command line.	Ctrl-E or End
Delete characters	Delete the character before the cursor.	Ctrl-H, Delete, or Backspace
	Delete the character following the cursor.	Ctrl-D
	Delete all characters from the cursor to the end of the line.	Ctrl-K
	Delete the whole line.	Ctrl-U or Ctrl-X
	Delete the word before the cursor.	Ctrl-W, Esc-Backspace, or Alt-Backspace
	Delete the word after the cursor.	Esc-D or Alt-D
Insert recently deleted text	Insert the most recently deleted text at the cursor.	Ctrl-Y
Display previous command lines	Scroll backward through the list of recently executed commands.	Ctrl-P or Up Arrow
	Scroll forward through the list of recently executed commands.	Ctrl-N or Down Arrow
	Search the command history in reverse order.	Ctrl-R
	Show list.	
Capitalization	Capitalize the word at the cursor; that is, make the first character uppercase and the rest of the word lowercase.	Esc-C

Category	Action	Keyboard Sequence
	Change the word at the cursor to all lowercase.	Esc-l
Special cases	Cancel a command; that is, clear a line.	Ctrl-C
	Quote insert character; that is, do not treat the next keystroke as an edit command.	Ctrl-V/Esc-Q
	Redraw the screen.	Ctrl-l
	Transpose characters.	Ctrl-T
	Enter multiline values when prompted for a value in the CLI (not available when editing a CLI command).	Esc-M
	Exit configuration mode.	Ctrl-Z

Filter Command Output

You can filter the output from a command by adding the pipe (|) symbol at the end of the command, followed by one of the filtering commands listed in the following table. You can chain together a series of filters on a single command line.

Table 2:

Filter	Description
append *filename*	Append output text to a file.
begin *regular-expression*	Begin with the line that matches a regular expression.
best-effort	Display data even if the data provider is unavailable, or continue loading from a file even if failures are occurring.
count	Count the number of lines in the output.
csv	Display the outfield fields in a comma-separated format.
display	Display the output as XML.
exclude *regular-expression*	Exclude lines that match a regular expression.
include *regular-expression*	Include lines that match a regular expression.
linnum	Enumerate lines in the output.
match-all	All selected filters must match.
match-any	At least one selected filter must match.
more	Paginate the output.
nomore	Suppress pagination of the output.

Filter	Description
notab	Display each output field on a separate line instead of in a table.
repeat *seconds*	Execute the command repeatedly, every specified number of seconds.
save *filename*	Save the output to a file.
select	For tabular output, select the columns to display.
tab	Enforce the table output of fields.
until *regular-expression*	End the display with the line that matches a regular expression.

Use Regular Expressions

The regular expressions available for use in filtering commands are a subset of those used in the UNIX **egrep** command and in the AWK programming language. The following table lists some common operators.

Table 3:

Operator	Action
.	Match any character.
^	Match the beginning of a string.
$	Match the end of a string.
[abc...]	Character class, which matches any of the characters abc... Character ranges are specified by a pair of characters separated by a -.
[^abc...]	Negated character class, which matches any character except abc.
r1 | r2	Alternation. It matches either r1 or r2.
r1r2	Concatenation. It matches r1 and then r2.
r+	Match one or more *r*s.
r*	Match zero or more *r*s.
r?	Match zero or one *r*s.
(r)	Grouping. It matches *r*.

Understand CLI Messages

The CLI displays messages at various times, such as when you enter and exit configuration mode, commit a configuration, and type a command or value that is not valid.

When you type an invalid command or value, a CLI message indicates the nature of the error:

```
vsmart# show c
Possible completions:
  certificate     Display installed certificate properties
```

```
  cli             Display cli settings
  clock           System clock
  configuration   Display configuration history
  control         Display Control Information
```

When you commit a configuration, the CLI first validates the configuration. If there is a problem, the CLI indicates the nature of the problem:

```
Entering configuration mode terminal
vsmart(config)# no vpn 0
vsmart(config)# commit
Aborted: 'vpn' : Cannot delete vpn 0
vsmart(config>)#
```

Count the Number of Lines in Command Output

To count the number of lines in the output from a command, use the **count** filtering command. For example:

```
vsmart# show interface | count
Count: 17 lines
```

Display Line Numbers in Command Output

To display line numbers in the output, use the **linnum** command filter. For example:

```
vsmart# show interface | linnum
1: interface vpn 0 interface eth0
2:  ip-address      10.0.1.12/24
3:  if-admin-status Up
4:  if-oper-status  Up
5:  encap-type      null
6:  mtu             1500
7:  hwaddr          00:50:56:00:01:02
8:  speed-mbps      1000
9:  duplex          full
10:  rx-packets      3035
11:  tx-packets      1949
12: interface vpn 0 interface eth1
13:  if-admin-status Down
14:  if-oper-status  Down
15:  hwaddr          00:0c:29:81:00:17
16:  rx-packets      0
17:  tx-packets      0
```

Search for a String in Command Output

To have the command output include only lines matching a regular expression, use the **include** command filter. For example:

```
vsmart# show cli | include screen
screen-length         30
screen-width          80
```

To have the command output include only the lines not containing a regular expression, use the **exclude** filtering command. For example:

```
vsmart# show cli | exclude e
history               100
prompt1               \h\M#
prompt2               \h\(m)#
```

To display the output starting at the first match of a regular expression, use the **begin** command filter. For example:

```
vsmart# show cli | begin show
show-defaults         false
terminal              linux
timestamp             disable
```

To end the command output when a line matches a regular expression, use the **until** command filter. For example:

```
vsmart# show cli | until history
autowizard            false
complete-on-space     true
history               100
```

Save Command Output to a File

To save command output to a file, use the **save** *filename* or **append** *filename* command filter. For example:

```
vsmart# show running-config omp | save filename
```

To save the configuration except for any passwords, add the **exclude password** command filter:

```
vsmart# show running-config system | exclude password | save filename
```

Configure a Device from the CLI

To configure a vSmart controller or vEdge router directly from the device, enter configuration mode:

```
vsmart# config
```

Then type either the full configuration command or type one command at a time to move down through the command hierarchy. Here is an example of typing a full configuration command:

```
vsmart(config)# vpn 1 interface ge0/1 ip address 1.1.1.1/16
```

Here is an example of moving down the command hierarchy by typing one command at a time:

```
vsmart(config)# vpn1
vsmart(config-vpn-1)# interface eth1
vsmart(config-interface-eth1)# ip address 1.1.1.1/16
vsmart(config-interface-eth1)#
```

To move to another portion of the hierarchy, simply type the name of the top-level command. For example:

```
vsmart(config-interface-eth1)# policy
vsmart(config-policy)#
```

To look at the configuration changes:

```
vsmart(config-policy)# top show configuration
vpn 1
 interface eth1
  ip address 1.1.1.1/16
  shutdown
 !
!
```

To commit the changes:

```
vsmart(config-policy)# commit
Commit complete.
```

Add Comments in a Configuration

All characters following an exclamation point (!) character up to the next newline in a configuration are ignored. This allows you to include comments in a file containing CLI commands and then paste the file into

the CLI. To enter the ! character as an argument or to include it in a password, prefix it with a backslash (\) or place it inside quotation marks (" ").

Delete Commands from a Configuration

Use the **no** command to delete commands from a configuration. For example:

```
vsmart(config)# do show running-config
vpn 1
 interface eth1
  ip address 1.1.1.1/16
  auto-negotiation
  shudown
  no proxy-arp
 !
!
vsmart(config)# no vpn 1 interface eth1 ip address
vsmart(config)# commit
commit complete.
vsmart(config)# do show running-config
vpn 1
 interface eth1
  auto-negotiation
  shudown
  no proxy-arp
 !
!
```