Chapter 3

Command-Line Interface

The command-line interface (CLI) is one tool for controlling the switch and displaying information about its status and configuration. This chapter describes the use of the CLI.

This chapter includes these sections:

- Section 3.1: Accessing the EOS CLI
- Section 3.2: Processing Commands
- Section 3.3: Kernel-based Virtual Machine Commands and Configuration
- Section 3.4: Switch Platforms
- Section 3.5: Command Modes
- Section 3.6: Managing Switch Configuration Settings
- Section 3.7: Other Command-Line Interfaces
- Section 3.8: Directory Structure
- Section 3.9: Command-Line Interface Commands

3.1 Accessing the EOS CLI

You can open an EOS CLI session through these connections:

- Ethernet management ports
- console port
- Telnet connections
- Secure Shell (SSH)
3.2 Processing Commands

3.2.1 Command Execution

Command keywords are not case-sensitive. The CLI also accepts truncated keywords that uniquely correspond to one command.

- The command abbreviation `con` does not execute a command in Privileged EXEC mode because the names of two commands begin with these letters: `configure` and `connect`.

  ```
  switch#con
  % Ambiguous command
  ```

- The command abbreviation `conf` executes `configure` in Privileged EXEC mode because no other command name begins with `conf`.

  ```
  switch#conf
  switch(config)#
  ```
3.2.2 Alias

The alias command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command.

Example

- This command makes srie an alias for the command show running-config interface ethernet 1-5.

```
switch(config)#alias srie show running-config interface ethernet 1-5
switch(config)#srie
interface Ethernet1
    switchport access vlan 33
    storm-control broadcast level 1
    spanning-tree portfast
    spanning-tree bpduguard enable
interface Ethernet2
    switchport access vlan 33
    spanning-tree portfast
interface Ethernet3
    switchport access vlan 33
    spanning-tree portfast
    spanning-tree bpduguard enable
interface Ethernet4
interface Ethernet5
    shutdown
```

3.2.3 Cursor Movement Keystrokes

EOS supports these cursor movement keystrokes:

- **Ctrl-B** or the **Left Arrow** key: moves cursor to the left.
- **Ctrl-F** or the **Right Arrow** key: moves cursor to the right.
- **Ctrl-A**: moves cursor to beginning of line.
- **Ctrl-E**: moves cursor to end of line.
- **Esc-B**: moves cursor left one word.
- **Esc-F**: moves cursor right one word.

3.2.4 History Substitution Keystrokes

The history buffer retains the last 20 commands entered. History substitution keystrokes that access previously entered commands include:

- **Ctrl-P** or the **Up Arrow** key: Recalls the most recent buffered commands. Repeat to recall older commands.
- **Ctrl-N** or the **Down Arrow** key: Recalls more recent commands after using the **Ctrl-P** or the **Up Arrow**. Repeat to recall newer commands.

The **show history** command in Privileged EXEC mode displays the history buffer contents.

```
switch#show history
en
config
exit
show history
```
### 3.2.5 Command Lists and Syntax Assistance

EOS CLI uses widely followed conventions for providing command lists and syntax assistance. These conventions are available in all command modes.

- To display all commands available at this level, type a question mark (?):

```plaintext
switchName>?
  clear      Reset functions
  connect    Open a terminal connection
  default    Set a command to its defaults
  disable    Turn off privileged commands
  enable     Turn on privileged commands
  exit       Exit from the EXEC
  logout     Exit from the EXEC
  no         Negate a command or set its defaults
  ping       Send echo messages
  show       Show running system information
  ssh        Open ssh connection
  tcpdump    Monitor packets with tcpdump
  telnet     Open a telnet connection
  terminal   Configure the terminal
  traceroute Trace route to destination
  watch      Execute a command repeatedly
  who        Display information about terminal lines
  zerotouch  ZeroTouch configuration
```

- To display a list of commands beginning with a specific character sequence, type the sequence followed by a question mark.

  ```plaintext
  switch#?
  di
  dir
  disable
  ```

- To display a command’s keywords or arguments, type a question mark as an argument.

  ```plaintext
  switch>?
  ping ?
  WORD  Ping destination address or hostname
  ip    IPv4 echo
  ipv6  IPv6 echo
  mpls  Send echo messages for LSP
  vrf   Ping in a VRF
  ```

- The switch accepts an address-mask or CIDR notation (address-prefix) in commands that require an IP address and mask. For example, these commands are processed identically:

  ```plaintext
  switch(config)#ip route 0.0.0.0 255.255.255.255 10.1.1.254

  switch(config)#ip route 0.0.0.0/32 10.1.1.254
  ```

- The switch accepts an address-wildcard or CIDR notation in commands requiring an IP address and wildcard. Wildcards use zeros to mask portions of the IP address and are found in some protocol configuration statements, including OSPF. The switch processes these commands identically:

  ```plaintext
  switch(config-router-ospf)#network 10.255.255.1 0.0.0.255 area 15

  switch(config-router-ospf)#network 10.255.255.1/24 area 15
  ```

### 3.2.6 Regular Expressions

A regular expression is a search pattern composed of symbols, letters and numbers. Some CLI parameters are defined as regular expressions for specifying more expressive search criteria. The switch uses regular expression pattern matching in several BGP commands.
The functionality of a regular expression for an AS-Path varies based on BGP regex asn and string mode configurations in the `ip as-path regex-mode` command.

Table 3-1 and Table 3-2 describe the behavior of special characters in asn and string modes respectively.

### Table 3-1 Functionality of Special Characters in ASN Mode

<table>
<thead>
<tr>
<th>Special Characters</th>
<th>Characters Names</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| .                  | Period           | Matches any AS number. | . matches ‘200’.  
10.20’ matches ‘10 30 20’, but does not match ‘10 20’. |
| ^                  | caret            | Matches the specified expression at the beginning of an input string. Also used to exclude expressions in brackets while matching. | ^123 matches ‘123’, ‘123 456’, ‘123 456 789’, and so on; but does not match ‘1234’.  
| *                  | Asterisk         | Matches an entire AS number that appears either zero or more times. | 200_100*, 300’ matches ‘200 300’, ‘200 100 300’, ‘200 100 100 300’, and so on.  
^100*$ matches empty AS path, ‘100’, ‘100 100’, ‘100 100 100’, and so on. |
| +                  | Plus sign        | Matches an entire AS number appearing either one or more times. | 10_20+, 30’ matches ‘10 20 30’, ‘10 20 20 30’ and so on; but does not match ‘10 200 30’. |
| $                  | Dollar sign      | Matches the specified expression at the end of an input string. | 1_2_3$ matches ‘1 2 3’, but does not match ‘1 2 3 4’. |
| []                 | Brackets         | Matches either an AS number, or a range of AS numbers separated by a hyphen. | [10_20, 30-39]’ matches ‘10’, ‘20’, ‘30’, ‘31’,...39’. |
| ?                  | Question mark    | Matches either zero or one occurrence of the pattern but the previous operand or entire AS number may appear zero or one time. | ‘100_200?’ matches ‘100’ and ‘100 200’.  
‘100_200?$’ does not match with ‘100 20’. |
| I                  | Pipe             | Matches the specified AS number on either side of the vertical bar. | 6400/6500’ matches either ‘6400’ or ‘6500’. |
| ( )                | Parenthesis      | Nests specified AS numbers for matching. | ^\(100(200|300))\$‘ matches either ‘100 200’ or ‘100 300’.  
^100_200|300_400\$‘ matches AS path either “100 200” or “300 400”. |
| _                  | Underscore       | Matches specified AS numbers that are converted into AS number delimiters. | ^_123_456_’ matches ‘123 456’.  
‘_333_444_’ matches ‘111 222 (333 444)’. |

**Note**

Precede the question mark (?) with Ctrl+V sequence to prevent it from being interpreted as a help command.
### Table 3-2 Functionality of Special Characters in String Mode

<table>
<thead>
<tr>
<th>Special Characters</th>
<th>Characters Names</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Period</td>
<td>Matches any single character.</td>
<td>‘1.2’ matches ‘102’.</td>
</tr>
<tr>
<td>^</td>
<td>Caret</td>
<td>Matches the specified expression at the beginning of an input string.</td>
<td>‘^123’ matches ‘123’, ‘1234’, ‘12345’, and so on. It also matches ‘123 456’, ‘123 456 789’, and so on.</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>Matches either zero or more sequences of the expression preceding the asterisk.</td>
<td>‘^5*$’ matches an empty AS path, ‘5’, ‘55’, ‘555’, and so on.</td>
</tr>
<tr>
<td>+</td>
<td>Plus sign</td>
<td>Matches either one or more sequences of the expression preceding the plus sign.</td>
<td>‘5+’ matches to ‘5’, ‘55’, ‘555’, and so on.</td>
</tr>
<tr>
<td>$</td>
<td>Dollar sign</td>
<td>Matches the expression at the end of an input string.</td>
<td>‘123$’ matches ‘123’, but does not match ‘1234’.</td>
</tr>
<tr>
<td>[]</td>
<td>Brackets</td>
<td>Matches either characters or a range of characters separated by a hyphen, within left and right brackets.</td>
<td>[025-7] matches ‘0’, ‘2’, and digits from ‘5’ to ‘7’; but does not match digits from ‘1’, ‘3’, ‘4’, ‘8’, and ‘9’.</td>
</tr>
<tr>
<td>?</td>
<td>Question mark</td>
<td>Matches either zero or one occurrence of the pattern.</td>
<td>‘12?3’ matches ‘13’ and ‘123’.</td>
</tr>
<tr>
<td>\</td>
<td>Pipe</td>
<td>Matches either one of the expressions or expression patterns on either side of the vertical bar.</td>
<td>‘14(36</td>
</tr>
<tr>
<td>()</td>
<td>Parenthesis</td>
<td>Nests specified expressions for matching.</td>
<td>‘(17)’ matches any number of the two-character string ’17’.</td>
</tr>
<tr>
<td>_</td>
<td>Underscore</td>
<td>For AS-Path regex, ‘_’ matches left brackets ‘([</td>
<td>, right brackets ’)]’), the beginning of input string, the end of input string, or space.</td>
</tr>
<tr>
<td>{}</td>
<td>Braces</td>
<td>Matches repetitions of the previous expression with the number of repetitions provided in braces.</td>
<td>‘10{2,3}’ matches ‘100’ and ‘1000’.</td>
</tr>
<tr>
<td>\</td>
<td>Backslash</td>
<td>Matches the character following the backslash and special characters.</td>
<td>‘(42’ matches ‘42’.</td>
</tr>
</tbody>
</table>

**Note**

Precede the question mark (?) with Ctrl+V sequence to prevent it from being interpreted as a help command.

### 3.2.7 Scheduling CLI Commands

The **schedule** command facilitates the periodic execution of the specified CLI command. Command parameters configure the time to start script execution, the interval between consecutive execution instances, the maximum time to execute the script, and the maximum number of files log that needs to be created.

The **schedule config** command sets configuration parameters to the CLI scheduler.
Chapter 3: Command-Line Interface

Processing Commands

The **show schedule** command lists the commands currently scheduled for periodic execution and displays the summary of the specified scheduled command.

**Examples**

- This command schedules the execution of a script once every 12 hours and the script execution is terminated if it exceeds 40 minutes. When max-log-files is set to zero, the script output is not logged.
  
  ```
  switch# schedule ms_1 interval 720 timeout 40 max-log-files 0 command bash /mnt/flash/myscript.sh
  ```

- This command saves the running configuration contents to a log file every hour, terminates the script execution if it exceeds 30 minutes and creates up to 24 log files.
  
  ```
  switch# schedule backup-test interval 60 max-log-files 24 command show running-config
  ```

- This command allows the switch to concurrently execute up to 2 scheduled commands.
  
  ```
  switch(config)# schedule config max-concurrent-jobs 2
  ```

- This command lists the commands that are scheduled for periodic execution.
  
  ```
  switch(config)# schedule config max-concurrent-jobs 3
  ```

  ```
  switch(config)# show schedule summary
  Maximum concurrent jobs  3
  Prepend host name to logfile: No
  Name            At time      Last    Interval   Timeout   Max     Logfile Location               Status
  ------------- ------------- ------- ---------- -------- -------- ------------------------------- ------
  tech-support      now        00:29      60        30      100     flash:schedule/tech-support/   Success
  thelp           12:02:00     00:02      60        40      100     flash:schedule/thelp/          Fail
  06/05/2018
  ```

3.2.8 Running Bash Shell Commands Automatically with Event Handlers

Event handlers execute a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after delay seconds.

To create an event handler, use the **event-handler** command. This creates a new event handler and places the CLI in event handler configuration mode for that handler. Use the **action bash** command to configure a Bash command to run when the handler is triggered, and the **trigger** command to specify the trigger. Event handlers can be triggered by various events, including:

- system booting
- a change in a specified interface’s operational status or IP address
- a change in the **startup-config** file
- a state change in a virtual machine monitored by VM Tracer

To change the delay period between the trigger and the action, use the **delay** command.

When an action is run, certain information is passed to it through environment variables. For the **boot** trigger, no variables are set. For the **interface** triggers, the following variables are set and passed to the action:

- **$INTF** interface name
- **$OPERSTATE** current operational status of the specified interface
- **$IP-PRIMARY** current primary IP address of the specified interface
To execute more than one Bash command in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the `action bash` command.

To display information about all event handlers or about a specific event handler, use the `show event-handler` command.

The `no event-handler` command deletes an event handler.

**Examples**

- These commands create an event handler named “eth_4” which will send email to a specified address when there is a change in the operational status of Ethernet interface 4:

```bash
switch(config)#event-handler eth_4
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#trigger on-intf ethernet 4 operstatus
switch(config-event-eth_4)#delay 60
switch(config-event-eth_4)#exit
switch(config)#
```

The above handler uses the `$OPERSTATE` variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.

- These commands create an event handler named “onStartup” which will execute a user-defined script 60 seconds after the system boots.

```bash
switch(config)#event-handler onStartup
switch(config-event-onStartup)#action bash /mnt/flash/startupScript1
switch(config-event-onStartup)#trigger onboot
switch(config-event-onStartup)#delay 60
switch(config-event-onStartup)#exit
switch(config)#
```

The above handler will also be executed on exiting from event-handler configuration mode.

- This command displays information about all event handlers configured on the system.

```bash
switch#show event-handler
Event-handler onStartup
Trigger: onBoot delay 60 seconds
Action: /mnt/flash/startupScript1
Last Trigger Activation Time: 1 minutes 51 seconds ago
Total Trigger Activations: 1
Last Action Time: 51 seconds ago
Total Actions: 1
switch#
```

- This command deletes the event handler named “onStartup”.

```bash
switch(config)#no event-handler onStartup
switch(config)#
```

**3.2.9 Running Adverse Drop Counters Monitor with Event Handlers**

A monitoring capability for adverse drop counters can be used as a warning that the system is encountering an abnormal condition. The adverse drop counter monitor runs periodically (with a default of 60 seconds) and performs the following actions:

- Reads the values of adverse drop counters.
- Compares each value to the value read in the previous run.
Chapter 3: Command-Line Interface Processing Commands

- If counter values increase more than a certain threshold (with a default of 100), it is considered as a threshold violation.
- If any counter has more than a certain number of threshold violations within a specific time window (with a default of 3 violations within 15 minutes) a syslog message is logged.

No configuration is required to enable adverse drop counters monitor with event handlers. It is enabled by default and can be disabled, and can be customized for duration of time window and threshold levels. To customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler, use the event-handler DropCountersHandler command. To display details of this event handler, use the show event-handler DropCountersHandler command.

Examples

- These commands customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.

  switch(config)#event-handler DropCountersHandler
  switch(config-DropCountersHandler)#action bash DropCounterLog.py -l
  switch(config-DropCountersHandler)#delay 0
  switch(config-DropCountersHandler)#trigger on-counters
  switch(config-DropCountersHandler-counters)#poll interval 60
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -w 800" > 0
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -c 5" > 0
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -t 200" > 0

- This command disables this event handler.

  switch(config)#no event-handler DropCountersHandler
  switch(config)#

- This command displays details of this event-handler.

  switch(config)#show event-handler DropCountersHandler
  Event-handler DropCountersHandler (BUILT-IN)
  Trigger: on-counters delay 0 seconds
  Polling Interval: 60 seconds
  Condition: bashCmd."DropCounterMonitor.py" > 0
  Threshold Time Window: 0 Seconds, Event Count: 1 times
  Action: DropCounterLog.py -l
  Action expected to finish in less than 20 seconds
  Total Polls: 39
  Last Trigger Detection Time: 38 minutes 22 seconds ago
  Total Trigger Detections: 1
  Last Trigger Activation Time: 38 minutes 22 seconds ago
  Total Trigger Activations: 1
  Last Action Time: Never
  Total Actions: 1

  switch(config)#
3.3 Kernel-based Virtual Machine Commands and Configuration

Arista’s EOS has leveraged its unmodified Linux kernel, and embraced open source standards-based technology that has brought operating system virtualization to Ethernet switching, utilizing the kernel-based virtual machine (KVM) as follows:

- The hypervisor is the Linux kernel.
- The core virtualization infrastructure is provided by the kernel module.
- The CPU-specific implementation is provided by the processor-specific module (Intel or AMD).
- The generic machine emulator and virtualizer KVM is provided by a Modified Quick Emulator (QEMU), which transforms the Linux kernel into the hypervisor.

The standard Linux kernel is the hypervisor, resulting in changes to the standard kernel (such as memory support and scheduler). Optimizations to these Linux components (such as a new scheduler in the 2.6 kernel) benefit both the hypervisor (host operating system) and Linux guest operating systems. With the kernel acting as the hypervisor, the switch can run other operating systems, such as Windows or Linux.

All components required are pre-installed with the Arista EOS software image, requiring only the download of the image. A few additional configuration steps get the KVM fully operational.

This chapter contains the following sections:

- **Section 3.3.1: KVM Commands**
- **Section 3.3.2: KVM Configuration**

### 3.3.1 KVM Commands

The following table covers KVM commands used throughout the configuration.

**Table 3-3 KVM Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>Up to 240 character comment for this mode.</td>
</tr>
<tr>
<td>default</td>
<td>Set a command to its defaults.</td>
</tr>
<tr>
<td>disk-image</td>
<td>Add Virtual Machine disk image.</td>
</tr>
<tr>
<td>enable</td>
<td>Enable VM.</td>
</tr>
<tr>
<td>exit</td>
<td>Exit from Virtual Machine configuration mode.</td>
</tr>
<tr>
<td>memory-size</td>
<td>Set memory size.</td>
</tr>
<tr>
<td>no</td>
<td>Negate a command or set its defaults.</td>
</tr>
<tr>
<td>show</td>
<td>Show running system information.</td>
</tr>
<tr>
<td>virtual-nic</td>
<td>Add virtual NIC.</td>
</tr>
<tr>
<td>vnc-port</td>
<td>Set VNC server port.</td>
</tr>
<tr>
<td>! !</td>
<td>Append to comment</td>
</tr>
</tbody>
</table>

### 3.3.1.1 CLI Commands

The following KVN CLI commands are used throughout the configuration.
vm
In config mode, the vm CLI command creates or deletes a KVM configuration, or enters config-vm mode. A newly created KVM will have an empty config file path and is disabled.

The CLI command syntax is as follows:

\[ \text{[no]} \text{ vm } \text{NAME} \]

Note
Deleting an enabled KVM first disables it, using the same process as the no enabled command in the config-vm mode.

config-file
In config-vm mode, the config-file CLI command sets the path of the libvirt config file, using standard file syntax (e.g. flash:vm/NetscalerVPX.xml or sata1:vm/NetscalerVPX.xml or /mnt/sata1/vm/NetscalerVPX.xml). Changing this value does not affect the state of a currently enabled KVM. To use the new file, the user must disable and then re-enable the KVM.

The CLI command syntax is as follows:

\[ \text{config-file } \text{[PATH]} \]

Note
If the file does not exist, a warning is printed and the new value is stored.

enabled
In config-vm mode, the enabled CLI command allows enabling a currently disabled VM, causing it to start up immediately. If a VM is enabled in the startup-config, it starts up automatically when EOS boots (or when VirtAgent starts).

The CLI command syntax is as follows:

\[ \text{[no]} \text{ enabled} \]

Disabling a currently enabled VM initiates a shutdown process in the following sequence:

- Attempt to shut down the VM politely if the guest OS supports ACPI.
- If the VM is still running after 30 seconds, terminate it.

show vm
In enable mode, the show vm CLI command prints information about the configuration and status of a KVM, or of all KVMs if NAME is omitted, as follows:

- Configuration:
  Name, config file path, and enabled.
- Status:
  PID, log file path, and serial console pty path.
- Current resource usage:
  RES, CPU%
- (Detailed only) contents of the config file.
- (Detailed only) contents of the log file.

The CLI command syntax is as follows:

\[ \text{show vm } \text{[detailed]} \text{ [NAME]} \]
attach vm
In enable mode, the attach vm CLI command connects to a KVM's serial console pty (using virsh console).

Note Press Ctrl-] to exit to the CLI.

The CLI command syntax is as follows:

```
attach vm [NAME]
```

show tech-support
The CLI command syntax is as follows:

```
show tech-support [detailed] [NAME]
```

reload
In enable mode, the reload CLI command is executed before restarting the system, and will shut down currently enabled KVMs using the same process as the no enabled command in config-vm mode.

The CLI command syntax is as follows:

```
reload
```

3.3.2 KVM Configuration
Arista EOS enables kernel-based virtual machine (KVM) instances by running KVM on the control-plane CPU of the switch. KVM instances can be defined from the CLI.

To configure a KVM, you must download the virtual machine image and configure the EOS.

This section contains the following topics:

- Section 3.3.2.1: Configuring a KVM
- Section 3.3.2.2: Configuring a Guest KVM

3.3.2.1 Configuring a KVM
To configure a KVM, perform the following steps:

**Step 1** Download the Virtual Machine Image to /mnt/flash

**Step 2** Name the virtual machine:

```
switch(config)#virtual-machine [kvm_name]
```

Example:

```
switch(config)#virtual-machine foo
```

**Step 3** Provide a pointer to the image:

```
switch(config-vm-foo)#disk-image [file:[path] image-format [format]
```

Example:

```
disk-image file:/mnt/flash/fedora.img image-format qcow2
```

**Step 4** Define the amount of memory allocated:

```
switch(config-vm-foo)#memory-size [size in bytes]
```

**Step 5** Bind the virtual NIC to an SVI (or management interface):

```
switch(config-vm-foo)#virtual-nic 1 vlan [1-4]
switch(config-vm-foo)#virtual-nic 1 management [1-4]
```
Step 6  Create the VNC server’s tcp port (display):
    switch(config-vm-foo)#vnc-port [vnc-port number]

Step 7  Enable the virtual machine:
    switch(config-vm-foo)#enable

Optionally attach to the virtual machine via VNC client pointed to the switch’s IP address. However, if Kernel hair-pinning is currently not enabled, preventing communication directly with the local switch, all traffic must have a destination on another networked device (such as a router, switch, or server).

For specifics about KVM please visit http://www.linux-kvm.org/.

Note
In the Real VNC Viewer for Options, Expert, and ColorLevel, if the default value is pal8, establishing a session may fail. If this occurs, set this value to full and reconnect.

Example
    switch#copy http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-openstack-sda.qcow2 flash:

    switch(config)#virtual-machine foo
    switch(config-vm-foo)#disk-image file:/mnt/flash/fedora.img image-format qcow2
    switch(config-vm-foo)#memory-size 512
    switch(config-vm-foo)#virtual-nic 1 vlan 1
    switch(config-vm-foo)#virtual-nic 2 management 1
    switch(config-vm-foo)#vnc-port 5900
    switch(config-vm-foo)#enable

3.3.2.2 Configuring a Guest KVM

To configure a guest KVM, perform the following steps:

Step 1  Download the Virtual Machine Image to /mnt/flash

Step 2  Name the virtual machine:
    switch(config)#virtual-machine [guest_name]

Example:
    switch(config)#virtual-machine guest123

Step 3  Provide a pointer to the image:
    switch(config-vm-guest123)#disk-image [file:[path] image-format [format]

Example:
    switch(config-vm-guest123)#disk-image flash:f16-x86_64-openstack-sda.qcow2 image-format ?
    iso iso image format
    qcow qcow image format
    qcow2 qcow2 image format
    raw raw image format
    vmdk vmdk image format
    switch(config-vm-guest123)#disk-image flash:f16-x86_64-openstack-sda.qcow2 image-format qcow2

Step 4  Define the amount of memory allocated:
    switch(config-vm-guest123)#memory-size [size in bytes]
Step 5  Bind the virtual NIC to an SVI (or management interface):
switch(config-vm-guest123)#virtual-nic 1 vlan [1-4]
switch(config-vm-guest123)#virtual-nic 2 management [1-4]

Step 6  Create the VNC server's tcp port (display):
switch(config-vm-guest123)#vnc-port [vnc-port number]

Step 7  Enable the virtual machine:
switch(config-vm-guest123)#enable
Example

```
switch#copy http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-openstack-sda.qcow2
(http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-openstack-sda.qcow2) flash:
...
switch#config terminal
switch(config)#virtual-machine ?
  WORD Virtual Machine name
switch(config)#virtual-machine foo
switch(config-vm-foo)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format ?
  iso     iso image format
  qcow    qcow image format
  qcow2   qcow2 image format
  raw     raw image format
  vmdk    vmdk image format
switch(config-vm-foo)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format qcow2
switch(config-vm-foo)#memory-size 1024
switch(config-vm-foo)#virtual-nic ?
  <1-4>        Virtual NIC Id
switch(config-vm-foo)#virtual-nic 1 ?
   Management    Management interface
   Vlan          Vlan interface
switch(config-vm-foo)#virtual-nic 1 vlan 1
switch(config-vm-foo)#virtual-nic 2 management 1
switch(config-vm-foo)#enable
switch(config-vm-foo)#^
switch#write mem
switch#show virtual-machine detail
Virtual Machine: foo
   Enabled:       Yes
   State:         Running
   Disk Image:    /mnt/flash/f16-x86_64-openstack-sda.qcow2
   Disk Image Format: qcow2
   Memory Size:   1024MB
   VNC port:      5900
   Virtual Nic:   vnic1
      Mac Address: 52:54:00:ee:11:c9
      Device:      Vlan1
      Model Type:  e1000
   Virtual Nic:   vnic2
      Mac Address: 52:54:00:df:2a:e1
      Device:      Management1
      Model Type:  e1000
```

Note

Once a Guest KVM has its configuration setup correctly, it can have virtual NIC connections in VLANs (inband), or on out-of-band management interfaces.
3.4 Switch Platforms

Features and CLI commands vary by switch platform. CLI options may also vary by switch platform for commands that are available on all platforms. Command descriptions in this manual describe feature availability and command parameters on the basis of switch platform, noting exceptions that exist among models that use a common platform.


These sections describe the following topics:

- Section 3.4.1: Viewing the Model Number
- Section 3.4.2: Determining a Switch's Operating Platform
- Section 3.4.3: Modular System Platforms – 7500 and 7500E Series Switches
- Section 3.4.4: Viewing Modules on 7300 Series Modular Switches
- Section 3.4.5: Multi-Chip Devices

3.4.1 Viewing the Model Number

To view the switch’s model number through the CLI, enter `show version`.

**Example**

- This command displays the model number, serial number, system MAC address, and manufacturing information of a DCS-7150S-64 switch.

  ```
  switch>show version
  Arista DCS-7150S-64-CL-F
  Hardware version: 01.01
  Serial number: JPE13120819
  System MAC address: 001c.7326.fd0c
  Software image version: 4.13.2F
  Architecture: i386
  Internal build version: 4.13.2F-1649184.4132F.2
  Internal build ID: eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1
  Uptime: 16 hours and 39 minutes
  Total memory: 4017088 kB
  Free memory: 1348228 kB
  switch>
  ```

3.4.2 Determining a Switch’s Operating Platform

**FM6000 Platforms**

To determine the operating platform on switch, display `platform` command options from Global Configuration command mode.
This command displays the operating platform of a switch operating on the FM6000 platform (7150 Series switches).

```
switch(config)#platform ?
  fm6000  FM6000 chip

switch(config)#platform
```

**Arad and Petra Platforms**

The `platform ?` command displays the same options on Arad and Petra platform switches. Refer to Section 3.4.1 to determine the switch’s model number.

- Fixed system switches (DCS-7048 Series) operate on the Petra platform.
- Modular switches (DCS-7500 Series) operate on Arad and Petra platforms. Section 3.4.3: Modular System Platforms – 7500 and 7500E Series Switches describe platform usage on these switches.

Arad and Petra platform switch typically utilize multiple chips. Section 3.4.5 describe methods of determining the port distribution on multi-chip platforms.

**Example**

- These commands display platform options of a switch operating on either Petra or Arad platforms.

```
switch(config)#platform ?
  arad    Arad switch chip
  fe1600  Fe1600 chip
  fe600   Fe600 fabric chip
  petraA  PetraA switch chip
  ptp     Precision Time Protocol
  sand    Sand platform

switch(config)#platform
```

**Trident and Trident-II Platforms**

The `platform ?` command returns `trident` on switches that operate on Trident or Trident-II platforms. Trident-II platform switches include options that configure the forwarding and routing tables. To determine the Trident platform that a switch uses, display `platform trident` options.

- These commands indicate that the switch is operating on the Trident-II platform:

```
switch(config)#platform ?
  ptp      Precision Time Protocol
  trident  Trident chip

switch(config)#platform trident ?
  fabric   Fabric configuration
  forwarding-table Forwarding table configuration
  mmu      Trident MMU configuration
  routing-table Routing table configuration

switch(config)#platform trident
```

Fixed and Modular switches are available that operate on the Trident-II platform. Refer to Section 3.4.1 to determine the switch’s model number. Section 3.4.4: Viewing Modules on 7300 Series Modular Switches displays the modules on a Trident-II platform modular switch.

Trident-II platform switches typically utilize multiple chips. Section 3.4.5 describe methods of determining port distribution on multi-chip platforms.
3.4.3 Modular System Platforms – 7500 and 7500E Series Switches

Modular switch platforms depend on their installed modules along with the fabric and forwarding software modes. The `show module` command displays the fabric modules in the switch. System performance in switches containing both module types is based on first-generation fabric capabilities. Best practice is to avoid switch configurations with mixed fabric modules.

These sections describe modular switch components and software modes that program their capacities.

3.4.3.1 Fabric Modules and Fabric Mode – 7500 and 7500E Series Switches

Each modular switch fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

Fabric mode determines the switch’s fabric performance capabilities. This mode must match the fabric modules in the switch. Fabric mode settings include:


E-series fabric modules can operate in `fe600` mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in `fe1600` mode. Switches containing both types of modules must be set to `fe600` mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

**Step 1** Switches reloading in `petraA` forwarding compatibility mode (Section 3.4.3.2) also reload in `fe600` fabric mode.

**Step 2** As specified by the `platform sand fabric mode (7500 and 7500E Series)` statement in `running-config`.

**Step 3** The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in `fe600` mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

**Example**

- This command configures the switch to reload in `fe1600` fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility

+-----------------+----------+
| Configuration   | Status   |
|-----------------+----------|
| Forwarding mode | None     |
| Fabric mode     | Fe1600   |
+-----------------+----------+

switch#
```
3.4.3.2 Linecard Modules and Forwarding Compatibility Mode – 7500 and 7500E Series

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecard modules support all basic switch functions.
- E-Series linecard modules support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the switch’s performance capabilities when forwarding data between linecard interfaces. Forwarding compatibility mode settings include:

- **PetraA**: Supports first-generation linecard modules.
- **Arad**: Supports E-Series linecard modules.

Forwarding compatibility mode determines the operational capacity of installed linecards. Table 3-4 lists the affect of the forwarding compatibility mode on linecard module types.

Table 3-4  Linecard Module and Forwarding Mode Performance

<table>
<thead>
<tr>
<th>Linecard Module Type</th>
<th>Forwarding Compatibility Mode</th>
<th>Linecard Operating Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-generation</td>
<td>petraA</td>
<td>First-generation performance capacity.</td>
</tr>
<tr>
<td>First-generation</td>
<td>arad</td>
<td>Linecard is powered-down.</td>
</tr>
<tr>
<td>E-Series</td>
<td>petraA</td>
<td>First-generation performance capacity.</td>
</tr>
<tr>
<td>E-Series</td>
<td>arad</td>
<td>E-series performance capacity.</td>
</tr>
</tbody>
</table>

**Important!** Switches must contain E-Series fabric modules to operate at E-Series performance capacities.

The forwarding compatibility mode is configured by the **platform sand forwarding mode (7500 and 7500E Series)** command. This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards.

Without a **platform sand forwarding mode** command, forwarding compatibility mode is determined by the first linecard that is operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in **petraA** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **platform sand forwarding mode** command.

**Example**

- This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

```
switch(config)#platform sand forwarding mode arad
switch(config)#
```

3.4.3.3 Viewing Modules – 7500 and 7500E Series

The **show module** command displays the model number of all installed modules.
This command displays the modules of a 7504 switch that contains first-generation modules.

```
switch>show module
Module | Ports | Card Type | Model        | Serial No.
------- | ------ |----------- |--------------|------------
1       | 2     | DCS-7500 Series Supervisor Module | 7500-SUP | JSH11440327
2       | 1     | Standby supervisor                  | Unknown  | Unknown
3       | 48    | 48-port SFF+ 10GigE Linecard       | 7548S-LC | JSH10449938
4       | 48    | 48-port SFF+ 10GigE Linecard       | 7548S-LC | JSH11091247
5       | 48    | 48-port SFF+ 10GigE Linecard       | 7548S-LC | JSH11211614
6       | 48    | 48-port SFF+ 10GigE Linecard       | 7548S-LC | JSH11520288
Fabric1 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11451230
Fabric2 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11451210
Fabric3 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11410115
Fabric4 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11380318
Fabric5 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11340955
Fabric6 | 0     | DCS-7504 Fabric Module              | 7504-FM  | JSH11410128
```

```
Module | MAC addresses | Hw | Sw | Status
------- | ------------- |----|----|-------
1       | 00:1c:73:03:06:ac - 00:1c:73:03:06:ac | 07.06 | 4.12.1 | Active
2       |              | 06.00 | 4.12.1 | Standby
3       | 00:1c:73:03:80:44 - 00:1c:73:03:80:73 | 07.10 | Ok
4       | 00:1c:73:03:e4:34 - 00:1c:73:03:e4:63 | 07.30 | Ok
5       | 00:1c:73:12:0b:3f - 00:1c:73:12:0b:6e | 08.00 | Ok
6       | 00:1c:73:12:b6:3f - 00:1c:73:12:b6:6e | 05.03 | Ok
Fabric1 |              | 05.03 | Ok
Fabric2 |              | 05.02 | Ok
Fabric3 |              | 05.02 | Ok
Fabric4 |              | 05.02 | Ok
Fabric5 |              | 05.02 | Ok
Fabric6 |              | 05.02 | Ok
switch>
```
This command displays modules of a 7504 switch that contains E-Series modules.

```
switch> show module
```

<table>
<thead>
<tr>
<th>Module</th>
<th>Ports</th>
<th>Card Type</th>
<th>Model</th>
<th>Serial No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>DCS-7500E-SUP Supervisor Module</td>
<td>7500E-SUP</td>
<td>JAS13060306</td>
</tr>
<tr>
<td>3</td>
<td>72</td>
<td>48 port 10GbE SFP+ &amp; 2x100G Linecard</td>
<td>7500E-72S-LC</td>
<td>JAS12410019</td>
</tr>
<tr>
<td>4</td>
<td>72</td>
<td>48 port 10GbE SFP+ &amp; 2x100G Linecard</td>
<td>7500E-72S-LC</td>
<td>JPE13041458</td>
</tr>
<tr>
<td>5</td>
<td>72</td>
<td>48 port 10GbE SFP+ &amp; 2x100G Linecard</td>
<td>7500S-72S-LC</td>
<td>JAS12380089</td>
</tr>
<tr>
<td>Fabric1</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12370008</td>
</tr>
<tr>
<td>Fabric2</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12380012</td>
</tr>
<tr>
<td>Fabric3</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12370014</td>
</tr>
<tr>
<td>Fabric4</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12380008</td>
</tr>
<tr>
<td>Fabric5</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12380017</td>
</tr>
<tr>
<td>Fabric6</td>
<td>0</td>
<td>DCS-7504-E Fabric Module</td>
<td>7504E-FM</td>
<td>JAS12370009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>MAC addresses</th>
<th>Hw</th>
<th>Sw</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:1c:73:00:f4:cd - 00:1c:73:00:f4:ce</td>
<td>00.00</td>
<td>4.12.3</td>
<td>Active</td>
</tr>
<tr>
<td>3</td>
<td>00:1c:73:00:9c:7b - 00:1c:73:00:9c:c2</td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>00:1c:73:28:a0:57 - 00:1c:73:28:a0:9e</td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>00:1c:73:00:9a:cb - 00:1c:73:00:9b:12</td>
<td>02.07</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric1</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric2</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric3</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric4</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric5</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
<tr>
<td>Fabric6</td>
<td></td>
<td>00.00</td>
<td>Ok</td>
<td></td>
</tr>
</tbody>
</table>

switch>
### 3.4.4 Viewing Modules on 7300 Series Modular Switches

7300 Series Modular switches operate on Trident-II platform. The `show module` command displays the model number of all installed modules.

```
switch>show module
Module    Ports Card Type                            Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     Supervisor 7300X SSD                 DCS-7300-SUP-D  JAS13340024
3         128   32 port 40GbE QSFP+ LC               7300X-32Q-LC    JPE13440416
4         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310113
5         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13340033
6         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310103
Fabric1   0     7304X Fabric Module                  7304X-FM        JAS13320077
Fabric2   0     7304X Fabric Module                  7304X-FM        JAS13350043
Fabric3   0     7304X Fabric Module                  7304X-FM        JAS13350050
Fabric4   0     7304X Fabric Module                  7304X-FM        JAS13350056

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:36:4b:71 - 00:1c:73:36:4b:72  01.01   4.13.3F Active
3         00:1c:73:58:d4:68 - 00:1c:73:58:d4:87  03.04           Ok
4         00:1c:73:36:05:61 - 00:1c:73:36:05:94  02.02           Ok
5         00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14  02.03           Ok
6         00:1c:73:36:02:e1 - 00:1c:73:36:03:14  02.02           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok
switch>
```

### 3.4.5 Multi-Chip Devices

Trident-II, Petra, and Arad platform switches and linecards utilize multiple chips, with Ethernet ports evenly distributed among the chips. Creating multi-port data structures (including port channels) that include ports from multiple chips protects against the failure of an individual chip on a device.

The following sections describe methods of determining port distribution on various switch platforms.

**Petra Fixed Switches**

7048-Series switches are Petra platform devices that distribute ports among two PetraA chips. The `show platform petraA port-info routing` command displays the ports that are controlled by each chip.

**Example**

- This command displays the following Ethernet port distribution on a DCS-7048-T switch:
  - Petra0 chip controls Ethernet 1 through Ethernet 32
Petra1 chip controls Ethernet 33 through Ethernet 52

```
switch#show platform petraA port-info routing
Petra0 Port Routing Information:
=========================================================================
<table>
<thead>
<tr>
<th>intfName</th>
<th>port-id</th>
<th>port-id</th>
<th>intfType</th>
<th>portType</th>
<th>v4</th>
<th>v6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CpuTm</td>
<td>2</td>
<td>0</td>
<td>Cpu</td>
<td>Tm</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet1</td>
<td>29</td>
<td>2</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet2</td>
<td>30</td>
<td>3</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet31</td>
<td>59</td>
<td>32</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet32</td>
<td>60</td>
<td>33</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RawPetra0/70</td>
<td>2118</td>
<td>70</td>
<td>Recycling Raw</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
```

Petra1 Port Routing Information:
=========================================================================
<table>
<thead>
<tr>
<th>intfName</th>
<th>port-id</th>
<th>port-id</th>
<th>intfType</th>
<th>portType</th>
<th>v4</th>
<th>v6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CpuTm</td>
<td>2</td>
<td>0</td>
<td>Cpu</td>
<td>Tm</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet33</td>
<td>66</td>
<td>2</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet52</td>
<td>85</td>
<td>21</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L3SecondHop1Petra1</td>
<td>86</td>
<td>22</td>
<td>Recycling Ethernet</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RawPetra1/70</td>
<td>2118</td>
<td>70</td>
<td>Recycling Raw</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
```

Petra Modular Switches

Linecards on 7500-Series modular switches distribute Ethernet ports among multiple petraA chips. The `show platform petraA port-info routing` command displays the ports that are controlled by each chip on all PetraA linecards or on a single linecard.

Example

- This command displays the following Ethernet port distribution on linecard 4 of a DCS-7504 switch:
  - Petra4/0 chip controls Ethernet 4/1 through Ethernet 4/8
  - Petra4/1 chip controls Ethernet 4/9 through Ethernet 4/16
  - Petra4/2 chip controls Ethernet 4/17 through Ethernet 4/24
  - Petra4/3 chip controls Ethernet 4/25 through Ethernet 4/32
  - Petra4/4 chip controls Ethernet 4/33 through Ethernet 4/40
Petra4/5 chip controls Ethernet 4/41 through Ethernet 4/48

```sh
switch(s1)# show platform petra module 4 port-info routing
Petra4/0 Port Routing Information:
========================================================================
<table>
<thead>
<tr>
<th>sys</th>
<th>fap</th>
<th>intfName</th>
<th>port-id</th>
<th>port-id</th>
<th>intfType</th>
<th>portType</th>
<th>v4</th>
<th>v6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CpuTm</td>
<td>2</td>
<td>0</td>
<td>Cpu</td>
<td>Tm</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Ethernet4/1</td>
<td>221</td>
<td>2</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/2</td>
<td>222</td>
<td>3</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/3</td>
<td>223</td>
<td>4</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/4</td>
<td>224</td>
<td>5</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/5</td>
<td>225</td>
<td>6</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/6</td>
<td>226</td>
<td>7</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/7</td>
<td>227</td>
<td>8</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4/8</td>
<td>228</td>
<td>9</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>RawPetra4/0/70</td>
<td>2118</td>
<td>70</td>
<td>Recycling</td>
<td>Raw</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>
Petra4/1 Port Routing Information:
========================================================================
<table>
<thead>
<tr>
<th>sys</th>
<th>fap</th>
<th>intfName</th>
<th>port-id</th>
<th>port-id</th>
<th>intfType</th>
<th>portType</th>
<th>v4</th>
<th>v6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CpuTm</td>
<td>2</td>
<td>0</td>
<td>Cpu</td>
<td>Tm</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Ethernet4/9</td>
<td>253</td>
<td>2</td>
<td>Nif</td>
<td>Ethernet</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>
switch(s1)#
```

Arad Modular Switches

7500-E Series linecards distribute Ethernet ports among multiple Arad chips. The `show platform arad port-info routing` command displays the ports that are controlled by each chip on all Arad linecards.

Example

- This command displays the following Ethernet port distribution on the 7500E-72S-LC linecard that is inserted as module 3 in a DCS-7508E switch:

```
switch#show platform arad mapping
```

<table>
<thead>
<tr>
<th>Arad3/0</th>
<th>Port</th>
<th>SysPhyPort</th>
<th>Voq</th>
<th>(Fap,FapPort)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xlge Serdes</td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CpuTm</td>
<td>2</td>
<td>32</td>
<td>(0, 0)</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3/1</td>
<td>28</td>
<td>240</td>
<td>(0, 2)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/2</td>
<td>29</td>
<td>248</td>
<td>(0, 3)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/3</td>
<td>30</td>
<td>256</td>
<td>(0, 4)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/4</td>
<td>31</td>
<td>264</td>
<td>(0, 5)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/5</td>
<td>32</td>
<td>272</td>
<td>(0, 6)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/6</td>
<td>33</td>
<td>280</td>
<td>(0, 7)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/7</td>
<td>34</td>
<td>288</td>
<td>(0, 8)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/8</td>
<td>35</td>
<td>296</td>
<td>(0, 9)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/9</td>
<td>36</td>
<td>304</td>
<td>(0,10)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/10</td>
<td>37</td>
<td>312</td>
<td>(0,11)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/11</td>
<td>38</td>
<td>320</td>
<td>(0,12)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/12</td>
<td>39</td>
<td>328</td>
<td>(0,13)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/13</td>
<td>40</td>
<td>336</td>
<td>(0,14)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/14</td>
<td>41</td>
<td>344</td>
<td>(0,15)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/15</td>
<td>42</td>
<td>352</td>
<td>(0,16)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/16</td>
<td>43</td>
<td>360</td>
<td>(0,17)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/17</td>
<td>44</td>
<td>368</td>
<td>(0,18)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/18</td>
<td>45</td>
<td>376</td>
<td>(0,19)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/19</td>
<td>46</td>
<td>384</td>
<td>(0,20)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/20</td>
<td>47</td>
<td>392</td>
<td>(0,21)</td>
<td>n/a</td>
</tr>
</tbody>
</table>

| RawArad3/0/56    | 2104| 16848      | (0, 56)| n/a          |
| Xlge Serdes      |      |-------------|      |               |
|                  |      |-------------|      |               |

<table>
<thead>
<tr>
<th>Arad3/1</th>
<th>Port</th>
<th>SysPhyPort</th>
<th>Voq</th>
<th>(Fap,FapPort)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xlge Serdes</td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3/21</td>
<td>60</td>
<td>496</td>
<td>(1, 2)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/34</td>
<td>73</td>
<td>600</td>
<td>(1,15)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/49/1</td>
<td>74</td>
<td>608</td>
<td>(1,16)</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3/49/12</td>
<td>85</td>
<td>696</td>
<td>(1,27)</td>
<td>n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arad3/2</th>
<th>Port</th>
<th>SysPhyPort</th>
<th>Voq</th>
<th>(Fap,FapPort)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xlge Serdes</td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3/35</td>
<td>92</td>
<td>752</td>
<td>(2, 2)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/48</td>
<td>105</td>
<td>856</td>
<td>(2,15)</td>
<td>n/a</td>
</tr>
<tr>
<td>Ethernet3/50/1</td>
<td>106</td>
<td>864</td>
<td>(2,16)</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3/50/12</td>
<td>117</td>
<td>952</td>
<td>(2,27)</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Trident-II Fixed Switches
Trident-II platform devices distribute their ports among multiple Trident II chips. The `show platform trident system port` command displays the ports that are controlled by each chip.

Example
- This command displays the following Ethernet port distribution on a DCS-7250QX-64-F switch:
  - Trident 0 chip controls Ethernet 1/1 through Ethernet 16/4
  - Trident 1 chip controls Ethernet 17/1 through Ethernet 32/4
  - Trident 2 chip controls Ethernet 33/1 through Ethernet 48/4
  - Trident 3 chip controls Ethernet 49/1 through Ethernet 64/4

```
switch# show platform trident system port
```

<table>
<thead>
<tr>
<th>Intf</th>
<th>Chip</th>
<th>ModId</th>
<th>Logical</th>
<th>Physical</th>
<th>MMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet1/1</td>
<td>Linecard0/0</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td>Ethernet1/2</td>
<td>Linecard0/0</td>
<td>1</td>
<td>2</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>Ethernet16/3</td>
<td>Linecard0/0</td>
<td>1</td>
<td>60</td>
<td>107</td>
<td>98</td>
</tr>
<tr>
<td>Ethernet16/4</td>
<td>Linecard0/0</td>
<td>1</td>
<td>61</td>
<td>108</td>
<td>99</td>
</tr>
<tr>
<td>Ethernet64/2</td>
<td>Linecard0/3</td>
<td>4</td>
<td>62</td>
<td>106</td>
<td>97</td>
</tr>
<tr>
<td>Ethernet64/3</td>
<td>Linecard0/3</td>
<td>4</td>
<td>63</td>
<td>107</td>
<td>98</td>
</tr>
<tr>
<td>Ethernet64/4</td>
<td>Linecard0/3</td>
<td>4</td>
<td>64</td>
<td>108</td>
<td>99</td>
</tr>
</tbody>
</table>

Trident-II Modular Switches
Linecards on 7300-Series modular switches distribute Ethernet ports among multiple Trident II chips. The `show platform trident system port` command can display the ports that are controlled by each chip on all linecards or on a single chip.

- This command displays the following Ethernet port distribution on DCS-7304-F switch that contains a 7300X-32Q-LC linecard as module 3:
  - Trident 0 chip controls Ethernet 1/1 through Ethernet 16/4 (on module 3)
• Trident 1 chip controls Ethernet 17/1 through Ethernet 32/4 (on module 3)

```bash
switch# show platform trident system port
<--------OUTPUT OMITTED FROM EXAMPLE-------->

<table>
<thead>
<tr>
<th>Port</th>
<th>Intf</th>
<th>Chip</th>
<th>ModId</th>
<th>Logical</th>
<th>Physical</th>
<th>MMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet3/1/1</td>
<td>Linecard3/0</td>
<td>5</td>
<td>1</td>
<td>17</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Ethernet3/2/1</td>
<td>Linecard3/0</td>
<td>5</td>
<td>2</td>
<td>21</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
<--------OUTPUT OMITTED FROM EXAMPLE-------->
| Ethernet3/16/3 | Linecard3/0 | 5         | 51    |        | 111      | 102 |
| Ethernet3/16/4 | Linecard3/0 | 5         | 52    |        | 112      | 103 |
<--------OUTPUT OMITTED FROM EXAMPLE-------->
| Ethernet3/32/3 | Linecard3/1 | 6         | 63    |        | 111      | 102 |
| Ethernet3/32/4 | Linecard3/1 | 6         | 64    |        | 112      | 103 |
<--------OUTPUT OMITTED FROM EXAMPLE-------->

switch#
3.5 Command Modes

Command modes define the user interface state. Each mode is associated with commands that perform a specific set of network configuration and monitoring tasks.

- **Section 3.5.1: Mode Types** lists the available modes.
- **Section 3.5.2: Navigating Through Command Modes** lists mode entry and exit commands.
- **Section 3.5.3: Command Mode Hierarchy** describes the mode structure.
- **Section 3.5.4: Group-Change Configuration Modes** describes editing aspects of these modes.

3.5.1 Mode Types

The switch includes these command modes:

- **EXEC**: EXEC mode commands display system information, perform basic tests, connect to remote devices, and change terminal settings. When logging into EOS, you enter EXEC mode.
  
  EXEC mode prompt: `switch>
  
- **Privileged EXEC**: Privileged EXEC mode commands configure operating and global parameters. The list of Privileged EXEC commands is a superset of the EXEC command set. You can configure EOS to require password access to enter Privileged EXEC from EXEC mode.
  
  Privileged EXEC mode prompt: `switch#
  
- **Global Configuration**: Global Configuration mode commands configure features that affect the entire system, such as system time or the switch name.
  
  Global Configuration mode prompt: `switch(config)#
  
- **Interface Configuration**: Interface configuration mode commands configure or enable Ethernet, VLAN, and Port-Channel interface features.
  
  Interface Configuration mode prompt: `switch(config-if-Et24)#
  
- **Protocol specific mode**: Protocol specific mode commands modify global protocol settings. Protocol specific mode examples include [ACL Configuration](#) and [Router BGP Configuration](#).
  
  The prompt indicates the active command mode. For example, the Router BGP command prompt is `switch(config-router-bgp)#`

3.5.2 Navigating Through Command Modes

To change the active command mode, perform one of these actions:

- To enter EXEC mode, log into the switch.
- To enter Privileged EXEC mode from EXEC, type `enable` (or `en`) followed, if prompted, by the enable password:
  
  `switch>en
  Password:
  switch#`

- To enter Global Configuration mode from Privileged EXEC, type `configure` (or `config`):
  
  `switch#config
  switch(config)#`

**Note**  
EOS supports `copy <url> running-config` in place of the `configure network` command.
• To enter Interface Configuration mode from Global Configuration, type interface and the name of
the interface to be modified:
  switch(config)# interface Et24
  switch(config-if-Et24)#

• To enter a protocol specific configuration mode from Global Configuration, type the required
command for the desired mode.
  switch(config)# router bgp 100
  switch(config-router-bgp)#

• To return one level from any configuration mode, type exit.
  switch(config)# exit
  switch#

• To return to Privileged EXEC mode from any configuration mode, type end or Ctrl-Z.
  switch(config-if-Et24)# <Ctrl-z>
  switch#

• To return to EXEC mode from Privileged EXEC mode, type disable (or dis).
  switch# dis
  switch>

• To exit EOS and log out of the CLI, type exit from EXEC mode or Privileged EXEC mode.
  switch# exit
  login:

3.5.3 Command Mode Hierarchy

Command modes are hierarchical. The parent mode of a specified command mode is the mode that
contains the command that enters the specified mode.

Example
  • EXEC mode contains the enable command, which enters Privileged EXEC mode. Therefore,
    EXEC is the parent mode of Privileged EXEC.

Commands that are executable in a specified command mode include all commands available in the
specified mode plus all commands executable from its parent mode.

Example
  • EXEC mode includes the ping command. EXEC mode is the parent mode of Privileged EXEC
    mode. Therefore, Privileged EXEC mode includes ping.

    Additionally, Privileged EXEC is the parent mode of Global Configuration mode. Therefore, Global
    Configuration mode also includes ping.

Executing a configuration mode command from a child mode may change the active command mode.

Example
  • Global Configuration mode contains interface ethernet and ip access-list commands, which
    enter Interface Configuration and Access Control List (ACL) Configuration modes, respectively.
    When the switch is in Interface Configuration mode, the ip access-list command is available and
    changes the active mode to ACL Configuration.
  switch(config)# interface ethernet 1
  switch(config-if-Et1)# ip access-list master-list
  switch(config-acl-master-list)#
The `exit` command changes the active command mode to its parent mode. When executed from Privileged EXEC or EXEC modes, the `exit` command terminates the session.

**Example**

- This command exits Global Configuration mode to Privileged EXEC mode.
  ```
  switch(config)#exit
  switch#
  ```

- This command terminates the user session.
  ```
  switch#exit
  ```

### 3.5.4 Group-Change Configuration Modes

Group-change modes apply all changes made during an edit session only after exiting the mode. Changes are stored when the user exits the mode, either through an `exit` or `end` command or through a command that enters a different configuration mode.

The `abort` command discards all changes not previously applied.

Access Control List (ACL) and Multiple Spanning Tree (MST) configuration modes are examples of group-change modes.
3.6 Managing Switch Configuration Settings

3.6.1 Verifying the Running Configuration Settings

*running-config* is the virtual file that stores the operating configuration. The *show running-config* command displays the *running-config*. The command is supported in Privileged EXEC mode.

**Example**

- Type *show running-config* in Privileged EXEC mode. The response in the example is truncated to display only the ip route configured.

```text
switch#show running-config
! Command: show running-config

<--------OUTPUT OMITTED FROM EXAMPLE-------->

! ip route 0.0.0.0/0 192.0.2.1
! <--------OUTPUT OMITTED FROM EXAMPLE-------->

end
switch#
```

3.6.2 Verifying Settings for the Current Mode

To display only the lines of *running-config* that affect the current mode, use the *active* option of the *show (various configuration modes)* command. This command option is available in all configuration modes except global configuration.

**Example**

- Type *show active* to display the content of *running-config* that affects the current mode. To include default settings in the display, type *show active all*.

```text
switch(config-router-ospf3)#show active all
ipv6 router ospf 9
    router-id 0.0.0.0
    default-metric 10
    distance ospf intra-area 10
    area 0.0.0.200 default-cost 10
    area 0.0.0.200
    no log-adjacency-changes
    timers spf 5
switch(config-router-ospf3)#
```

To display any comments associated with the current mode, use the *comment* option of the *show (various configuration modes)* command.

**Example**

- Type *show comment* to display any comments attached to the current mode.

```text
switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
switch(config-router-ospf3)#
```
3.6.3 Adding a Comment to a Configuration Mode

To add a comment to most switch configuration modes, use the `comment (various configuration modes)` command. Comments cannot be modified, but can be replaced by entering the `comment` command again and entering new text. Comments cannot be added to global configuration mode.

To append to an existing comment, enter `!!` followed by additional comment text. To display comments for the active mode, use the `show comment` command. The `no comment` and `default comment` commands remove the comment from `running-config`.

**Examples**

- These commands enter a comment in Router OSPF3 Mode.

  ```
  switch(config-router-ospf3)#comment
  Enter TEXT message. Type 'EOF' on its own line to end.
  Consult Thomas Morton before making changes to the OSPF configuration.
  EOF
  switch(config-router-ospf3)#
  ```

- These commands append additional information to the comment entered above.

  ```
  switch(config-router-ospf3)#!! x2735
  switch(config-router-ospf3)#show comment
  Comment for router-ospf3:
  Consult Thomas Morton before making changes to the OSPF configuration.
  x2735
  switch(config-router-ospf3)#
  ```

3.6.4 Saving the Running Configuration Settings

`startup-config` is the file, stored in internal flash memory, that the switch loads when it boots. Configuration changes that are not saved to `startup-config` are lost the next time the switch is booted.

The `write` and `copy running-config startup-config` commands store the operating configuration to `startup-config`. Both commands are supported in Privileged EXEC mode.

**Example**

- These equivalent commands save the current operating configure to the startup-config file.

  ```
  switch#write
  ```

  ```
  switch#copy running-config startup-config
  ```

The `show startup-config` command displays the startup configuration file. The command is supported in Privileged EXEC mode.
Example

- Type `show startup-config` to display the startup configuration file. The response in the example is truncated to display only the ip route configured in Admin Username.

```
switch# show startup-config
! Command: show startup-config
! Startup-config last modified at Wed Feb 19 08:34:31 2014 by admin
!

<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip route 0.0.0.0/0 192.0.2.1
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch#
```
3.7 Other Command-Line Interfaces

EOS can access other CLIs that provide switch commands, files, and services.

- Section 3.7.1: Aboot Command-Line Interface describes the boot-loader CLI
- Section 3.7.2: Bash Shell describes the Bash shell CLI.

3.7.1 Aboot Command-Line Interface

Aboot is the switch boot loader. It reads a configuration file from the internal flash or a USB flash drive and attempts to boot a software image. The switch opens an Aboot shell if the switch does not find a software image, the configuration is corrupted, or the user terminates the boot process. The Aboot shell provides a CLI for manually booting a software image, recovering the internal flash to its default factory state, running hardware diagnostics, and managing files.

3.7.2 Bash Shell

The switch provides a Linux Bash shell for accessing the underlying Linux operating system and extensions. The Bash shell is accessible in all command modes except EXEC. Section 3.5.1: Mode Types describes EOC command modes.

- To enter the Bash, type `bash` at the prompt.
  ```bash
  switch#bash
  Arista Networks EOS shell
  [admin@Switch ~]$ 
  ```

- To exit the Bash, type `logout`, `exit`, or Ctrl-D at the Bash prompt.
  ```bash
  [admin@Switch ~]$ logout
  switch#
  ```
3.8 Directory Structure

EOS operates from a flash drive root mounted as the `/mnt/flash` directory on the switch. The EOS CLI supports these file and directory commands:

- **delete**: Delete a file or directory tree.
- **copy**: Copy a file.
- **more**: Display the file contents.
- **diff**: Compares the contents of files located at specified URLs.
- **rename**: Rename a file
- **cd**: Change the current working directory.
- **dir**: Lists directory contents, including files and subdirectories.
- **mkdir**: Create a directory.
- **rmdir**: Remove a directory.
- **pwd**: Display the current working directory.

Verify flash memory space before copying a file. When a file is copied to flash, it is first written to a temporary file and then renamed to the destination rather than directly overwriting the destination file. This protects the integrity of the existing file if the `copy` command is interrupted, but requires more free space to complete the process.

Switch directory files are accessible through the Bash shell and Aboot. When entering the Bash shell from the switch, the working directory is located in `~/home` and has the name of the user name from which Bash was entered.

**Example**

- These commands were entered from the user name john:

  ```
  switch# bash
  [john@switch ~]$ pwd
  /home/john
  [john@switch ~]$
  ```

  In this instance, the working directory is `/home/john`

When a flash drive is inserted in the USB flash port, flash drive contents are accessible through `/mnt/usb1`.

When entering Aboot, the working directory is the root directory of the boot.
3.9 Command-Line Interface Commands

Mode Navigation Commands
- alias
- bash
- configure (configure terminal)
- configure network
- copy running-config
- daemon
- disable
- enable
- end
- exit

File Transfer Commands
- ip ftp client source-interface
- ip http client local-interface
- ip ssh client source-interface
- ip tftp client source-interface

File Management Commands
- copy running-config
- dir
- pwd

Modular Switch Platform Commands
- platform arad lag mode
- platform arad lag mode
- platform sand fabric mode (7500 and 7500E Series)
- platform sand forwarding mode (7500 and 7500E Series)
- platform sand lag hardware-only
- show platform sand compatibility
- show platform sand lag hardware-only

CLI Scheduling Commands
- schedule
- schedule config
- show schedule

Event Handler Commands
- action bash
- delay
- event-handler
- event-handler DropCountersHandler
- show event-handler
- show event-handler DropCountersHandler
- trigger

Terminal Parameter Commands
- terminal length
- terminal monitor
Display and Comment Commands

- comment (various configuration modes)
- show (various configuration modes)
- show module
- show version
action bash

The action bash command specifies a Bash shell command to be run when an event handler is triggered. When an event handler is triggered, execution of the associated shell command is delayed by a configurable period set by the delay command. Only a single Bash command may be configured for an event handler, but the command may have multiple arguments. If more than one Bash command must be executed in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the action bash command.

To specify the event that will trigger the action, use the trigger command.

If the event handler uses an on-intf trigger, the following environment variables are passed to the action and can be used as arguments to the Bash command:

- $INF interface name.
- $OPERSTATE current operational status of the specified interface.
- $IP-PRIMARY current primary IP address of the specified interface.

Command Mode
Event-Handler Configuration

Command Syntax

action bash command

Parameters

- command Bash shell command to be executed when the event handler is triggered.

Example

- This command configures the event handler “onStartup” to run a script on the flash drive.

  switch(config-handler-onStartup)#action bash /mnt/flash/myScript1

- This command configures the event handler “eth_4” to send email to the specified address when there is a change in the operational status of Ethernet interface 4.

  switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"

The above action uses the $OPERSTATE variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.
alias

The alias command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command. Once created, an alias is accessible in all modes and all user sessions, but is subject to all the restrictions of the original command.

When using a command alias, no tokens may precede the alias except the no and default keywords. However, an alias can incorporate positional parameters.

In online help, aliases are preceded by an asterisk (*) in this format:

*alias_name=command_name

The no alias and default alias commands remove the specified alias.

Command Mode
Global Configuration

Command Syntax

alias alias_name command_name
no alias alias_name
default alias alias_name

Parameters

- **alias_name** the string which is to be substituted for the original command. The string can include letters, numbers, and punctuation, but no spaces. If the **alias_name** string is identical to an existing command, the alias will supersede the original command.

- **command_name** the command which is to be executed when the alias is entered in the CLI. If the original command requires additional parameters, they must be included in the **command_name** string in the following manner:

  Positional parameters are of the form "%n" and must be whitespace-delimited. The first parameter is represented by “%1” and any additional parameters must be numbered sequentially. When executing the alias a value must be entered for each parameter or the CLI will display the error “% incomplete command”.

Examples

- This command makes e an alias for the command enable.
  switch(config)#alias e enable

- This command makes srie an alias for the command show running-config interface ethernet 1-6.
  switch(config)#alias srie show running-config interface ethernet 1-6

- These commands make ss an alias for the command show interfaces ethernet <range> status with a positional parameter for the port range, then use the alias to display the status of ports 4/1-4/5.
  switch(config)#alias ss show interfaces ethernet %1 status
  switch(config)#ss 4/1-4/5

<table>
<thead>
<tr>
<th>Port</th>
<th>Name</th>
<th>Status</th>
<th>Vlan</th>
<th>Duplex</th>
<th>Speed</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Et4/1</td>
<td></td>
<td>connected</td>
<td>in Po1</td>
<td>full</td>
<td>10000</td>
<td>10GBASE-SRL</td>
</tr>
<tr>
<td>Et4/2</td>
<td></td>
<td>notconnect</td>
<td>in Po1</td>
<td>full</td>
<td>10000</td>
<td>10GBASE-SRL</td>
</tr>
<tr>
<td>Et4/3</td>
<td></td>
<td>notconnect</td>
<td>1</td>
<td>full</td>
<td>10000</td>
<td>10GBASE-SRL</td>
</tr>
<tr>
<td>Et4/4</td>
<td></td>
<td>notconnect</td>
<td>1</td>
<td>full</td>
<td>10000</td>
<td>10GBASE-SRL</td>
</tr>
<tr>
<td>Et4/5</td>
<td></td>
<td>notconnect</td>
<td>1</td>
<td>full</td>
<td>10000</td>
<td>10GBASE-SRL</td>
</tr>
</tbody>
</table>
bash

The bash command starts the Linux Bash shell. The Bash shell gives you access to the underlying Linux operating system and system extensions.

To exit the Bash, type logout, exit, or Ctrl-D at the Bash prompt.

**Command Mode**
- Privileged EXEC

**Command Syntax**

bash

**Examples**

- This command starts the Bash shell.
  ```
  switch#bash
  Arista Networks EOS shell
  [admin@switch ~]$ 
  ```

- This command, executed within Bash, exits the Bash shell.
  ```
  [admin@switch ~]$ logout
  switch#
  ```
comment (various configuration modes)

The `comment` command adds a comment for the active configuration mode to `running-config`. Comments cannot be modified, but can be replaced by entering the `comment` command again and entering new text. To append to an existing comment, enter `!!` followed by additional comment text. To display comments, use the `comment` option of the `show (various configuration modes)` command.

The `no comment` and `default comment` commands remove the comment from `running-config`. Comments cannot be added to the global configuration mode through the EOS.

**Command Mode**
All configuration modes except Global Configuration

**Command Syntax**
```
comment comment_text EOF
no comment
default comment
!! comment_text
```

**Parameters**
- `comment_text` To create a comment, enter a message when prompted. The message may span multiple lines.
- `EOF` To end a comment, type `EOF` on its own line (case sensitive) and press `enter`.

**Example**
- This command adds a comment to the active configuration mode.
  ```
  switch(config-sg-radius-RAD-SV1)#comment
  Enter TEXT message. Type 'EOF' on its own line to end.
  Consult Thomas Morton before making changes to the RADIUS configuration.
  EOF
  switch(config-sg-radius-RAD-SV1)#
  ```
- This command appends a line to the comment for the active configuration mode.
  ```
  switch(config-sg-radius-RAD-SV1)!! x3452
  switch(config-sg-radius-RAD-SV1)#
  ```
configure (configure terminal)

The `configure` command places the switch in the Global Configuration mode to configure features at the system level. You can move to Interface Configuration mode and protocol-specific mode from the Global Configuration mode. The command may also be entered as `configure terminal`.

**Command Mode**
Privileged EXEC

**Command Syntax**
```plaintext
configure [terminal]
```

**Example**
- This command places the switch in the Global Configuration mode.
  ```plaintext
  switch>enable
  switch#configure
  switch(config)#
  ```
configure checkpoint

The `configure checkpoint` command saves the running configuration to a checkpoint file. This checkpoint file can be used for restoring the current running configuration in future, if required.

**Command Mode**
- Privileged EXEC

**Command Syntax**
```
configure checkpoint {restore checkpoint_name | save [checkpoint_name]}
```

**Parameters**
- `restore checkpoint_name` restores the running configuration from the specified checkpoint file.
- `save checkpoint_name` saves running configuration to the specified checkpoint file.

**Guidelines**
If the filename already exists, EOS overwrites the filename. If the command is entered without a checkpoint name, the switch automatically saves the checkpoint under the name `ckp-date-number` where `date` is the date in YYYYMMDD format and `number` increments by one for each automatically named checkpoint file.

**Examples**
- This command saves `running-config` to the `ca_test` checkpoint file.
  ```
  switch# configure checkpoint save ca_test
  ```
- This command restores the `running-config` from the `ca_test` checkpoint file.
  ```
  switch# configure checkpoint restore ca_test
  ! Preserving static routes. Use 'no ip routing delete-static-routes' to clear them.
  ```
- This command saves `running-config` to the 13Aug2018 checkpoint file. The `dir` command shows the contents of the checkpoint directory.
  ```
  switch# configure checkpoint save
  switch# dir checkpoint:
  Directory of checkpoint:/
  -rw-  7426 Aug 13 12:00  ckp-20180813-17
  -rw-  7588 Aug 13 12:10  ckp-20180813-18
  -rw-  8499 Aug 13 12:13  ckp-20180813-19
  -rw-  8499 Aug 13 12:13  ckp-20180813-20
  ```
configure convert

The configure convert command converts the current configuration syntax to the specified syntax.

Command Mode
Privileged EXEC

Command Syntax
configure convert new-syntax

Parameter
• new-syntax converts running-config to the current version of EOS.

Example
• This command converts running-config to the current version of EOS.
  switch#configure convert new-syntax

WARNING!
Converting existing configuration to new syntax will lose backward compatibility.
Make sure you won't downgrade to releases that only support the old syntaxes.

Proceed [ y/n ]
configure network

The `configure network` command is deprecated. Use the `copy <url> running-config` command to configure the switch from a local file or network location.
copy running-config

The current operating configuration of the switch is stored in a virtual file called running-config. The copy running-config command saves the contents of the running-config virtual file to a new location.

Command Mode
Privileged EXEC

Command Syntax
  copy running-config DESTINATION

Parameters
- DESTINATION  destination for the contents of the running-config file. Values include:
  - startup-config  the configuration file that the switch loads when it boots.
  - file  a file in the switch file directory.
  - flash  a file in flash memory.
  - url  any valid URL.

The copy running-config url and write network url commands are equivalent.

Examples
- This command copies running-config to the startup-config file.
  switch#copy running-config startup-config
  switch#
- This command copies running-config to a file called rc20110617 in the dev subdirectory of the switch directory.
  switch#copy running-config file:dev/rc20110617
  switch#
daemon

The `daemon` command accesses daemon configuration mode for adding or removing external daemons and scripts, which are then managed by ProcMgr.

The `no daemon` and `default daemon` commands delete the daemon by removing the corresponding `daemon` command from `running-config`.

**Command Mode**
Global Configuration

**Command Syntax**

```
daemon daemon_name
no daemon daemon_name
default daemon daemon_name
```

**Parameters**
- `daemon_name` label that references the daemon configuration mode.

**Examples**
- These commands enters daemon configuration mode and initiates the daemon script.
  
  ```
  switch(config)#daemon process1
  switch(config-daemon-process1)#command process-script -i -m
  switch(config-daemon-process1)#
  ```
delay

The `delay` command specifies the time in seconds the system will delay between a triggering event and the execution of an event handler action. The default delay is 20 seconds.

**Command Mode**
Event-Handler Configuration

**Command Syntax**

```
delay seconds
```

**Parameters**
- `seconds` number of seconds to delay before executing the action. The default is 20.

**Example**
- This command configures the event handler Eth5 to delay 10 seconds before executing.

```
switch(config-handler-Eth5)#delay 10
switch(config-handler-Eth5)#
```
dir

The `dir` command displays a list of files on a file system.

**Command Mode**
Privileged EXEC

**Command Syntax**
`dir [SCOPE][FILE TYPE]`

**Parameters**
- **SCOPE** the files to display. Options include
  - `<no parameter>` lists normal files in current directory.
  - `/all` list all files, including hidden files
  - `/recursive` list files recursively
- **FILE TYPE** The options include:
  - `<no parameter>` lists undeleted files
  - `all_filesystems` list files on all filesystems including deleted files, undeleted files, and files with errors
  - `extensions` directory or file name
  - `file` directory or file name
  - `flash` directory or file name
  - `supervisor-peer` directory or file name
  - `system` directory or file name
  - `usb1` directory or file name

**Example**
- This command displays the flash directory.

```plaintext
switch# dir flash:
Directory of flash:/
-rwx  293409892  293409892 Oct 23 08:55  EOS-4.11.0.swi
-rwx  221274543  221274543 Sep 6 13:37  EOS-4.7.5.swi
-rwx  271453650  271453650 Sep 4 19:13  EOS_4.10.1-SSO.swi
-rwx     135168     135168 Dec 31  1979  FSCK0000.REC
-rwx         26         26 Oct 23 13:51  boot-config
-rwx        8570        8570 Sep 10 12:22  cfg_sso_mst
-rwx        5642        5642 Sep 20 10:35  config.reset
drwx        4096        4096 Oct 23 13:59  debug
-rwx         12         12 Oct 23 13:56  kernel-params
drwx        4096        4096 Oct 23 14:59  persist
drwx        4096        4096 Sep 6 14:50  schedule
-rwx        5970        5970 Oct 23 13:53  startup-config
switch#
```
**disable**

The `disable` command exchanges the session’s current command mode with the specified privilege level.

**Command Mode**

Privileged EXEC

**Command Syntax**

```
disable [PRIVILEGE_LEVEL]
```

**Parameters**

- `PRIVILEGE_LEVEL` Session’s new privilege level. Value ranges from 0 to 15. Levels 2 through 15 place the switch in Privileged EXEC mode. Values of 0 or 1 leave the switch in EXEC mode.
  - `<no parameter>` Session is assigned default level of 1.
  - `<0 to 15>` Specifies session level.

**Restrictions**

New privilege level must be less than the session’s current level.

**Examples**

- This command exits Privileged EXEC mode level of 15 to enter EXEC mode level 1.
  ```
  switch# disable
  switch>
  ```
enable

The `enable` command places the switch in Privileged EXEC mode. If an `enable` password is set, the CLI displays a password prompt when a user enters the `enable` command. If the user enters an incorrect password three times, the CLI displays the EXEC mode prompt.

To set a local `enable` password, use the `enable password` command.

Command Mode

EXEC

Command Syntax

```
enable [PRIVILEGE_LEVEL]
```

Parameters

- `PRIVILEGE_LEVEL` Session's privilege level. Values range from 0 to 15. Values of 0 or 1 places the switch in EXEC mode. Any level above 1 leaves the switch in Privileged EXEC mode.
  - `<no parameter>` Session is assigned default level of 15.
  - `<0 to 15>` Specifies session level.

Example

- This command places the switch in Privileged EXEC mode with the default privilege level of 15.
  ```
  switch>enable
  switch#
  ```
end

The end command exits to Privileged Exec mode from any Configuration mode. If the switch is in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the end command also saves all pending changes made in that mode to running-config.

**Command Mode**

All configuration modes

**Command Syntax**

end

**Example**

- This command exits to Privileged Exec mode.

```bash
switch(config-if-Et25)#end
switch#
```
event-handler

An event handler executes a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after delay seconds.

The event-handler command places the switch in event-handler configuration mode for the specified event handler. If the named event handler does not already exist, this command creates it. Event-handler configuration mode is a group change mode that configures event handlers.

Changes made in a group change mode are saved by leaving the mode through the exit command or by entering another configuration mode.

These commands are available in event-handler configuration mode:

- **action bash**
- **delay**
- **trigger**

The no event-handler and default event-handler commands delete the specified event handler by removing it from running config.

Command Mode

Global Configuration

Command Syntax

```
  event-handler name
  no event-handler name
  default event-handler name
```

Parameters

- **name**  name of the event handler to be configured. If the named event handler does not already exist, this command will create it.

Example

- This command places the switch in event-handler configuration mode for an event handler called “Eth_5”.

  ```
  switch(config)#event-handler Eth_5
  switch(config-handler-Eth_5)#
  ```
event-handler DropCountersHandler

The event-handler DropCountersHandler command enables the adverse drop counters monitor with event handlers. The DropCountersHandler event handler is enabled by default, and can be customized for duration of time window and threshold levels.

The no event-handler DropCountersHandler command disables the adverse drop counters monitor with event handlers. The default event-handler DropCountersHandler command resets the DropCountersHandler event handler to the system default.

Command Mode

Global Configuration

Command Syntax

- event-handler DropCountersHandler
- no event-handler DropCountersHandler
- default event-handler DropCountersHandler

Examples

- These commands customize the delay, polling interval, and condition for width (-w), violation count (-c), and threshold (-t) of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.
  
  switch(config)#event-handler DropCountersHandler
  switch(config-DropCountersHandler)#action bash DropCounterLog.py -l
  switch(config-DropCountersHandler)#delay 0
  switch(config-DropCountersHandler)#trigger on-counters
  switch(config-DropCountersHandler-counters)#poll interval 60
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -w 800" > 0
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -c 5" > 0
  switch(config-DropCountersHandler-counters)#condition bashCmd."DropCounterMonitor.py" -t 200" > 0
  
- This command disables this event handler.
  
  switch(config)#no event-handler DropCountersHandler
  switch(config)#
exit

The `exit` command places the switch in the parent of the command mode from which the `exit` command was entered.

- When used in Global configuration, the switch enters Privileged EXEC mode.
- When used in EXEC or Privileged EXEC mode, the `exit` command terminates the user session.
- When the command is used in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the `exit` command also applies all pending changes made in that mode.

**Command Mode**

All modes

**Command Syntax**

`exit`

**Example**

- This command exits Global Configuration mode to Privileged EXEC mode.
  
  ```
  switch(config)#exit
  switch#
  ```

- This command terminates the user session.
  
  ```
  switch#exit
  ```
**ip ftp client source-interface**

By default, the FTP (File Transfer Protocol) source IP address is selected by the switch (the IP address of the source interface if one is assigned). The `ip ftp client source-interface` command allows the user to override the default FTP source address.

The `ip ftp client source-interface` and `ip ftp source-interface` commands are functionally equivalent. In each case, `ip ftp client source-interface` is stored in `running-config`.

The `no ip ftp client source-interface` and `default ip ftp client source-interface` commands restore default behavior by removing the `ip ftp client source-interface` statement from `running-config`.

**Command Mode**

Global Configuration

**Command Syntax**

```
ip ftp [client] source-interface INTERFACE [vrf vrf_name]
no ip ftp [client] source-interface
default ip ftp [client] source-interface
```

**Parameters**

- **client** Parameter has no functional effect.
- **INTERFACE** Interface providing the IP address. Options include:
  - `ethernet e_num` Ethernet interface specified by `e_num`.
  - `loopback l_num` Loopback interface specified by `l_num`.
  - `management m_num` Management interface specified by `m_num`.
  - `port-channel p_num` Port-channel interface specified by `p_num`.
  - `tunnel t_num` Tunnel interface specified by `t_num`.
  - `vlan v_num` VLAN interface specified by `v_num`.
  - `vrf vrf_name` Uses the specified user-defined VRF.

**Examples**

- These commands configure the 10.10.121.15 as the source IP address the switch uses when communicating with FTP servers.
  ```
  switch(config)#interface ethernet 17
  switch(config-if-Et17)#ip address 10.10.121.15/24
  switch(config-if-Et17)#ip ftp client source-interface ethernet 17
  switch(config)#
  ```

- This command configures the switch to use interface tunnel 45 and vrf vrf01 when communicating with FTP servers.
  ```
  switch(config)#ip ftp client source-interface tunnel 45 vrf vrf01
  switch(config)#
  ```
ip http client local-interface

The `ip http client local-interface` command specifies the source IP address for hypertext transfer protocol (HTTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The `no ip http client local-interface` and `default ip http client local-interface` commands restore default behavior by removing the `ip http client local-interface` statement from `running-config`.

**Command Mode**
Global Configuration

**Command Syntax**

```
ip http client local-interface INTERFACE [vrf vrf_name]
no ip http client local-interface
default ip http client local-interface
```

**Parameters**

- **INTERFACE** Interface providing the IP address. Options include:
  - `ethernet e_num` Ethernet interface specified by `e_num`.
  - `loopback l_num` Loopback interface specified by `l_num`.
  - `management m_num` Management interface specified by `m_num`.
  - `port-channel p_num` Port-channel interface specified by `p_num`.
  - `vlan v_num` VLAN interface specified by `v_num`.
  - `vrf vrf_name` Uses the specified user-defined VRF.

**Examples**

- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with HTTP servers.
  ```
  switch(config)#interface vlan 10
  switch(config-if-Vl10)#ip address 10.15.17.9/24
  switch(config-if-Vl10)#ip http client local-interface vlan 10
  switch(config)#
  ```
- This command configures the switch to use interface tunnel 45 and vrf vrf01 when communicating with HTTP servers.
  ```
  switch(config)#ip http client local-interface tunnel 45 vrf vrf01
  switch(config)#
  ```
**ip ssh client source-interface**

The `ip ssh client source-interface` command specifies the source IP address for secure shell (SSH) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The `ip ssh client source-interface` and `ip ssh source-interface` commands are functionally equivalent. In each case, `ip ssh client source-interface` is stored in `running-config`.

The `no ip ssh client source-interface` and `default ip ssh client source-interface` commands restore default behavior by removing the `ip ssh client source-interface` statement from `running-config`.

**Command Mode**

Global Configuration

**Command Syntax**

```
ip ssh [client] source-interface INTERFACE [vrf vrf_name]
nip ssh [client] source-interface
default ip ssh [client] source-interface
```

**Parameters**

- `client` Parameter has no functional effect.
- `INTERFACE` Interface providing the IP address. Options include:
  - `ethernet e_num` Ethernet interface specified by `e_num`.
  - `loopback l_num` Loopback interface specified by `l_num`.
  - `management m_num` Management interface specified by `m_num`.
  - `port-channel p_num` Port-channel interface specified by `p_num`.
  - `vlan v_num` VLAN interface specified by `v_num`.
  - `vrf vrf_name` Uses the specified user-defined VRF.

**Examples**

- These commands configure the 10.17.17.9 as the source IP address the switch uses when communicating with SSH servers.
  ```
  switch(config)#interface vlan 10
  switch(config-if-Vl10)#ip address 10.17.17.9/24
  switch(config-if-Vl10)#ip ssh client source-interface vlan 10
  switch(config)#
  ```
- This command configures the switch to use interface tunnel 45 and vrf vrf01 when communicating with SSH servers.
  ```
  switch(config)#ip ssh client source-interface tunnel 45 vrf vrf01
  switch(config)#
  ```
**ip tftp client source-interface**

The `ip tftp client source-interface` command specifies the source IP address for Trivial File Transfer Protocol (TFTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The `ip tftp client source-interface` and `ip tftp source-interface` commands are functionally equivalent. In each case, `ip tftp client source-interface` is stored in *running-config*.

The `no ip tftp client source-interface` and `default ip tftp client source-interface` commands restore default behavior by removing the `ip tftp client source-interface` statement from *running-config*.

**Command Mode**
- Global Configuration

**Command Syntax**

```
ip tftp [client] source-interface INTERFACE [vrf vrf_name]
no ip tftp [client] source-interface
default ip tftp [client] source-interface
```

**Parameters**
- `client` Parameter has no functional effect.
- `INTERFACE` Interface providing the IP address. Options include:
  - `ethernet e_num` Ethernet interface specified by `e_num`.
  - `loopback l_num` Loopback interface specified by `l_num`.
  - `management m_num` Management interface specified by `m_num`.
  - `port-channel p_num` Port-channel interface specified by `p_num`.
  - `vlan v_num` VLAN interface specified by `v_num`.
  - `vrf vrf_name` Uses the specified user-defined VRF.

**Examples**
- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with TFTP servers.
  ```
  switch(config)#interface vlan 10
  switch(config-if-Vl10)#ip address 10.15.17.9/24
  switch(config-if-Vl10)#ip tftp client source-interface vlan 10
  switch(config)#
  ```
- This command configures the switch to use interface tunnel 45 and vrf vrf01 when communicating with TFTP servers.
  ```
  switch(config)#ip tftp client source-interface tunnel 45 vrf vrf01
  switch(config)#
  ```
**platform arad lag mode**

The **platform arad lag mode** command allows configuration of LAGs with more than 16 members.

**Command Mode**
Global Configuration

**Command Syntax**

```plaintext
platform arad lag mode [1024x16 | 256x64 | 512x32]
```

**Examples**

- This command configures 1024 LAGs with 16 members each.
  ```plaintext
  switch(config)# platform arad lag mode 1024x16
  ! Change will take effect only after switch reboot.
  switch(config)#
  ```

- This command configures 256 LAGs with 64 members each.
  ```plaintext
  switch(config)# platform arad lag mode 256x64
  ! Change will take effect only after switch reboot.
  switch(config)#
  ```

- This command configures 512 LAGs with 32 members each.
  ```plaintext
  switch(config)# platform arad lag mode 512x32
  ! Change will take effect only after switch reboot.
  switch(config)#
  ```
platform sand fabric mode (7500 and 7500E Series)

The **platform sand fabric mode** command specifies the fabric mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

The fabric mode determines the modular switch’s fabric performance capabilities and must be compatible with the installed fabric modules. Fabric mode settings include:

- **fe600**: Supports first-generation fabric modules.
- **fe1600**: Supports E-Series fabric modules.

**Important!** Switches that reload in **petraA** forwarding compatibility mode (**platform sand forwarding mode (7500 and 7500E Series)**) also reload in **fe600** fabric mode regardless of the presence of a **platform sand fabric mode** statement in **running-config**.

The switch’s fabric mode setting must match the capabilities of its installed fabric modules. Reloading the switch in a different mode may be required after exchanging fabric modules for a different module type. The **show module** command displays the fabric modules in the switch.

Each fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

E-series fabric modules can operate in **fe600** mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in **fe1600** mode. Switches containing both types of modules must be set to **fe600** mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

**Step 1** Switches reloading in **petraA** forwarding compatibility mode also reload in **fe600** fabric mode.

**Step 2** As specified by the **platform sand fabric mode** statement in **running-config**.

**Step 3** The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in **fe600** mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

The **no platform sand fabric mode** and **default platform sand fabric mode** commands remove the **platform sand fabric mode** command from **running-config**.

**Command Mode**

Global Configuration

**Command Syntax**

```
platform sand fabric mode [MODE_SETTING]
no platform sand fabric mode
default platform sand fabric mode
```

**Parameters**

- **MODE_SETTING** Specifies the switch’s fabric mode. Options include:
  - **fe1600**: E-Series fabric mode.
  - **fe600**: First-generation fabric mode.
Examples

- This command configures the switch to reload in **fe1600** fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility
```

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding mode</td>
<td>None</td>
</tr>
<tr>
<td>Fabric mode</td>
<td>Fe1600</td>
</tr>
</tbody>
</table>

switch#
platform sand forwarding mode (7500 and 7500E Series)

The **platform sand forwarding mode** command specifies the forwarding compatibility mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

Forwarding compatibility mode specifies switch forwarding capabilities and configures performance capacity of installed linecards. Forwarding compatibility modes settings include:

- **petraA**: Supports first-generation fabric modules.
- **ara**: Supports E-Series fabric modules.

**Important!** Switches that reload in **petraA** forwarding compatibility mode also reload in **fe600** fabric mode regardless of the presence of a **platform sand fabric mode (7500 and 7500E Series)** statement in **running-config**.

This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards. The **show module** command displays the linecard modules in the switch.

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecards support all basic switch functions.
- E-Series linecards support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the operational capacity of installed linecards. **Table 3-5** lists the affect of the forwarding compatibility mode on all linecard module types.

### Table 3-5 Linecard Module and Forwarding Mode Performance

<table>
<thead>
<tr>
<th>Linecard Module Type</th>
<th>Forwarding Software Mode</th>
<th>Linecard Operating Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-generation</td>
<td>petraA</td>
<td>Linecard performs at first-generation performance capacity.</td>
</tr>
<tr>
<td>First-generation</td>
<td>ara</td>
<td>Linecard is powered-down.</td>
</tr>
<tr>
<td>E-Series</td>
<td>petraA</td>
<td>Linecard performs at first-generation performance capacity.</td>
</tr>
<tr>
<td>E-Series</td>
<td>ara</td>
<td>Linecard performs at E-series performance capacity.</td>
</tr>
</tbody>
</table>

**Important!** Linecards operate at E-Series performance capacities only on switches that contain E-Series fabric modules and have a fabric mode setting of **fe1600** fabric mode (**platform sand fabric mode (7500 and 7500E Series)**).

Without a **platform sand forwarding mode** command, forward compatibility mode is determined by the first linecard that becomes operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in **petraA** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **platform sand forwarding mode** command.

The **no platform sand forwarding mode** and **default platform sand forwarding mode** commands restore the **platform sand forwarding mode** command from **running-config**.

**Command Mode**

Global Configuration
Command Syntax

platform sand forwarding mode [MODE_SETTING]
no platform sand forwarding mode
default platform sand forwarding mode

Parameters

- **MODE_SETTING**  Specifies the switch’s software forwarding mode. Options include:
  - **arad**  the switch supports E-Series linecard capabilities.
  - **petraA**  the switch supports first-generation linecard capabilities.

Examples

- This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

  switch(config)#platform sand forwarding mode arad
  switch(config)#
platform sand lag hardware-only

The platform sand lag hardware-only command specifies that all LAGs will use hardware resources including single member LAGs. Hardware resource allocation and deallocation traffic disruption occurs on the first member addition or deletion, rather than the second member addition or deletion.

The no platform sand lag hardware-only and default platform sand lag hardware-only commands specify that LAGs are not required to be implemented in hardware, and therefore some LAGs may be implemented in software. Permitting both hardware and software LAGs may increase the total number of port-channels because we have no resource limit on the number of software LAGs.

Command Mode
   Global Configuration

Command Syntax
   platform sand lag hardware-only
   no platform sand lag hardware-only
   default platform sand lag hardware-only

Examples
   • This command configures all LAGs to use hardware resources. All existing one member LAGs will be allocated hardware resources, when available.

     switch(config)#platform sand lag hardware-only
     switch(config)#

   • This command allows certain LAGs (single member LAGs) to not consume hardware resources. All existing one member LAGs will release their hardware resources.

     switch(config)#no platform sand lag hardware-only
     switch(config)#
**pwd**

The `pwd` command displays the working directory.

**Command Mode**
Privileged EXEC

**Command Syntax**

```
pwd
```

**Examples**

- This command shows that the working is Flash.

```
switch# pwd
flash:/
switch#
```
**schedule**

The `schedule` command facilitates the periodic execution of a specified CLI command. Command parameters configure the start time of periodic execution, the interval between consecutive execution instances, the maximum time allotted for command execution, and the maximum number of log files that can be created.

The `no schedule` and `default schedule` commands disable execution of the specified command.

**Command Mode**

Global Configuration

**Command Syntax**

```
schedule schedule_name PERIOD {max-log-files count | timeout timeout_interval}
{command cmd | logging verbose | loglocation flash:}
no schedule schedule_name
default schedule schedule_name
```

**Parameters**

- **name**  
  label associated with the scheduled command.

- **PERIOD**  
  start time for execution and interval between consecutive execution instances. The interval ranges from 2 to 1440 minutes. The default interval while scheduling the `show tech-support` command is 60 minutes. Options include:
  - **at**  
    start time for execution. Options include:
    - `hh:mm:ss interval interval`  
      The command execution starts at the specified time and repeats at the specified interval.
    - `hh:mm:ss mm/dd/yyyy interval interval`  
      The command execution starts at the specified time on the specified day and repeats at the specified interval.
    - `hh:mm:ss once`  
      The command execution starts at the specified time and does not repeat.
    - `hh:mm:ss mm/dd/yyyy once`  
      The command execution starts at the specified time on the specified day and does not repeat.
    - `hh:mm:ss yyyy-mm-dd interval interval`  
      The command execution starts at the specified time on the specified day and repeats at the specified interval.
    - `hh:mm:ss yyyy-mm-dd once`  
      The command execution starts at the specified time on the specified day and does not repeat.
  - **interval interval**  
    The command execution starts immediately and repeats at the specified interval.
  - **now interval interval**  
    The command execution starts immediately and repeats at the specified interval.
  - **max-log-files count**  
    maximum number of log files command generates for command output. The count of maximum log files ranges from 1 to 10000. The default count of maximum log files while scheduling the `show tech-support` command is 100.
  - **timeout timeout_interval**  
    maximum time allotted for the script execution. The timeout interval ranges from 1 to 480 minutes. The default timeout is 30 minutes.

**Note**

The command execution is terminated if it exceeds the specified `timeout` interval. The timeout allotted for the scheduled command must not be greater than the corresponding interval.

- **command cmd**  
  The command that needs to be executed.
- **logging verbose**  Sets the logging level to “verbose.” A syslog entry is added after the execution of the scheduled command, regardless of whether the scheduled command has succeeded or failed. In the absence of **logging verbose**, the syslog entry is added only if the execution of the scheduled command fails with an error.

- **loglocation destination**  The flash destination for scheduled command output files.

**Guidelines**

Log files created by the command are stored in the `flash:/schedule/scheduled_name` directory. Empty log files are created for commands that do not generate any output.

**Examples**

- This command saves the running configuration contents to the log file every hour with immediate effect and creates a maximum of 24 log files.
  
  ```
  switch(config)# schedule backup-test interval 60 max-log-files 24 command show running-config
  ```

- This command starts the script execution at 12:00:00 and repeats every 720 minutes. The script execution is terminated if it exceeds 20 minutes. It generates a maximum of one log file because the specified bash command does not have an output.
  
  ```
  switch(config)# schedule ms1 at 12:00:00 interval 720 timeout 20 max-log-files 1 command bash /mnt/flash/myscript.sh
  ```

The **show schedule** command lists the commands currently scheduled for periodic execution and displays the summary of the specified scheduled command.

```
switch#show schedule summary
Maximum concurrent jobs 1
Prepend host name to logfile: Yes
Name    At time    Last  time  Interval  Timeout    Max   Logfile Location Status
    ---    -------    ------    ----    --------    -----    --------------------- -------
ms1    now     23:03    720    20     1         flash:/schedule/ms1   Success
switch#
```
schedule config

The schedule config command sets configuration parameters to the CLI scheduler.

The no schedule config max-concurrent-jobs and default schedule config max-concurrent-jobs commands reset the limit of maximum concurrent jobs to the default value of 1 by removing the corresponding schedule config max-concurrent-jobs statement from running-config.

The no schedule config prepend-hostname-logfile and default schedule config prepend-hostname-logfile commands reset the log filenames to the default state.

Command Mode
Global Configuration

Command Syntax

```
schedule config {max-concurrent-jobs limit | prepend-hostname-logfile}
no schedule config {max-concurrent-jobs limit | prepend-hostname-logfile}
default schedule config {max-concurrent-jobs limit | prepend-hostname-logfile}
```

Parameters

- **max-concurrent-job limit** specifies the maximum number of concurrent commands that can run on the switch. The maximum concurrent jobs ranges from 1 to 4. The default value is 1.
- **prepend-hostname-logfile** enables prepending hostnames to log filenames. By default, this option is enabled.

Examples

- This command configures to concurrently run a maximum of three commands on the switch.

  ```
switch(config)#schedule config max-concurrent-jobs 3
  switch(config)#show schedule summary
  Maximum concurrent jobs  3
  Prepend host name to logfile: No
  Name            At time      Last    Interval   Timeout   Max     Logfile Location               Status
  time     (mins)    (mins)    log
  files
  ------------- ------------- ------- ---------- -------- -------- ------------------------------- ------
  tech-support      now        00:29      60        30      100     flash:schedule/tech-support/   Success
  thelp           12:02:00     00:02      60        40      100     flash:schedule/thelp/          Fail
  06/05/2018
  switch(config)#
  ```

- This command enables prepending the hostname to log filenames.

  ```
switch(config)#schedule config prepend-hostname-logfile
  switch(config)#show schedule summary
  Maximum concurrent jobs  3
  Prepend host name to logfile: Yes
  Name            At time      Last    Interval   Timeout   Max     Logfile Location               Status
  time     (mins)    (mins)    log
  files
  ------------- ------------- ------- ---------- -------- -------- ------------------------------- ------
  tech-support      now        00:29      60        30      100     flash:schedule/tech-support/   Success
  thelp           12:02:00     00:02      60        40      100     flash:schedule/thelp/          Fail
  06/05/2018
  switch(config)##
show (various configuration modes)

The `show` command, when executed within a configuration mode, can display data in `running-config` for the active configuration mode.

**Command Mode**

All configuration modes except Global Configuration

**Command Syntax**

```
show [DATA_TYPE]
```

**Parameters**

- `DATA_TYPE` Specifies display contents. Values include:
  - `active` Displays `running-config` settings for the configuration mode.
  - `active all` Displays `running-config` plus defaults for the configuration mode.
  - `active all detail` Displays `running-config` plus defaults for the configuration mode.
  - `comment` Displays comment entered for the configuration mode.

**Related Commands**

The `show` commands in ACL-configuration mode and MST-configuration mode include the `active` and `comment` options along with additional mode-specific options.

**Example**

```
switch(config-sg-tacacs+-TAC-GR)#show active
server TAC-1
server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```
show event-handler

The **show event-handler** command displays the contents and activation history of a specified event handler or all event handlers.

**Command Mode**
- Privileged EXEC

**Command Syntax**

```
show event-handler [handler_name]
```

**Parameters**

- `handler_name` optional name of an event handler to display. If no parameter is entered, the command displays information for all event handlers configured on the system.

**Example**

- This command displays information about an event handler called “eth_5”.

```
switch# show event-handler Eth_5
Event-handler Eth_5
Trigger: on-intf Ethernet5 on ip delay 20 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: 
Device-health Action: None
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: 15 days 2 hours 19 minutes ago
Total Trigger Detections: 1
Last Trigger Activation Time: 15 days 2 hours 19 minutes ago
Total Trigger Activations: 1
Last Action Time: 15 days 2 hours 19 minutes ago
Total Actions: 1
switch#
```
show event-handler DropCountersHandler

The `show event-handler` command displays details of the DropCountersHandler event handler.

**Command Mode**
Privileged EXEC

**Command Syntax**
`show event-handler DropCountersHandler`

**Example**
- This command displays details of this event handler.

```plaintext
switch(config)#show event-handler DropCountersHandler
Event-handler DropCountersHandler (BUILT-IN)
  Trigger: on-counters delay 0 seconds
    Polling Interval: 60 seconds
    Condition: bashCmd."DropCounterMonitor.py" > 0
  Threshold Time Window: 0 Seconds, Event Count: 1 times
  Action: DropCounterLog.py -l
  Action expected to finish in less than 20 seconds
  Total Polls: 39
  Last Trigger Detection Time: 38 minutes 22 seconds ago
  Total Trigger Detections: 1
  Last Trigger Activation Time: 38 minutes 22 seconds ago
  Total Trigger Activations: 1
  Last Action Time: Never
  Total Actions: 1

switch(config)#
```
show module

The `show module` command displays information that identifies the supervisor, fabric, and linecard modules in a modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

**Command Mode**

EXEC

**Command Syntax**

```
show module [MODULE_NAME]
```

**Parameters**

- **MODULE_NAME** Specifies modules for which data is displayed. Options include:
  - <no parameter> All modules (identical to `all` option).
  - linecard `line_num` Linecard module. Number range varies with switch model.
  - supervisor `super_num` Supervisor module. Number range varies with switch model.
  - `mod_num` Supervisor (1 to 2) or linecard (3 to 18) module.
  - `all` All modules.

**Related Commands**

- `show version` displays model and serial numbers of modular system components.
Example
This command displays information about all installed modules on a DCS-7504 switch.

```
switch# show module
Module | Ports | Card Type | Model         | Serial No.   
-------+-------+-----------+---------------+--------------
  1    |   2   | DCS-7500 Series Supervisor Module | 7500-SUP | JSH11440327  
  2    |   1   | Standby supervisor          | Unknown  | Unknown      
  3    |   48  | 48-port SFP+ 10GigE Linecard | 7548S-LC | JSH10315938  
  4    |   48  | 48-port SFP+ 10GigE Linecard | 7548S-LC | JSH11665247  
  5    |   48  | 48-port SFP+ 10GigE Linecard | 7548S-LC | JSH11834614  
  6    |   48  | 48-port SFP+ 10GigE Linecard | 7548S-LC | JSH11060688  
Fabric1 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11244430  
Fabric2 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11892120  
Fabric3 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11941115  
Fabric4 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11661618  
Fabric5 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11757555  
Fabric6 | 0    | DCS-7504 Fabric Module     | 7504-FM  | JSH11847728  
Module | MAC addresses | Hw | Sw | Status
-------+-----------------+----+----+-------
  1    | 00:1c:23:03:06:ac - 00:1c:23:03:06:ac | 07.06 | 4.12.1 | Active  
  2    |                                            | 4.12.1 | Standby
  3    | 00:1c:23:03:08:44 - 00:1c:23:03:08:73 | 06.00 | Ok   
  4    | 00:1c:23:03:e4:34 - 00:1c:23:03:e4:63 | 07.10 | Ok   
  5    | 00:1c:23:12:0b:3f - 00:1c:23:12:0b:6e | 07.30 | Ok   
  6    | 00:1c:23:12:b6:3f - 00:1c:23:12:b6:6e | 08.00 | Ok   
Fabric1 |                            | 05.03 | Ok   
Fabric2 |                            | 05.03 | Ok   
Fabric3 |                            | 05.02 | Ok   
Fabric4 |                            | 05.02 | Ok   
Fabric5 |                            | 05.02 | Ok   
Fabric6 |                            | 05.02 | Ok   
switch#
```
This command displays information about all installed modules on a DCS-7304 switch.

```
switch# show module
Module | Ports | Card Type | Model            | Serial No.
-------|-------|-----------|------------------|-----------
1       | 3     | Supervisor 7300X SSD | DCS-7300-SUP-D | JAS13340024
3       | 128   | 32 port 40GbE QSFP+ LC | 7300X-32Q-LC | JPE13440416
4       | 64    | 48 port 10GbE SFP+ & 4 port QSFP+ LC | 7300X-64S-LC | JAS13310113
5       | 64    | 48 port 10GbE SFP+ & 4 port QSFP+ LC | 7300X-64S-LC | JAS13340033
6       | 64    | 48 port 10GbE SFP+ & 4 port QSFP+ LC | 7300X-64S-LC | JAS13310103
Fabric1 | 0     | 7304X Fabric Module | 7304X-FM | JAS13320077
Fabric2 | 0     | 7304X Fabric Module | 7304X-FM | JAS13350043
Fabric3 | 0     | 7304X Fabric Module | 7304X-FM | JAS13350050
Fabric4 | 0     | 7304X Fabric Module | 7304X-FM | JAS13350056
```

```
Module | MAC addresses | Hw | Sw  | Status
-------|---------------|----|-----|------
1       | 00:1c:73:36:4b:71 - 00:1c:73:36:4b:72 | 01.01 | 4.13.3F Active
3       | 00:1c:73:58:d4:68 - 00:1c:73:58:d4:87 | 03.04 | Ok
4       | 00:1c:73:36:05:61 - 00:1c:73:36:05:94 | 02.02 | Ok
5       | 00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14 | 02.03 | Ok
6       | 00:1c:73:36:02:e1 - 00:1c:73:36:03:14 | 02.02 | Ok
Fabric1 | 00.00 | Ok
Fabric2 | 00.00 | Ok
Fabric3 | 00.00 | Ok
Fabric4 | 00.00 | Ok
switch#
```
show platform sand compatibility

The `show sand platform compatibility` command displays the fabric and forwarding modes. These modes determine switch forwarding capabilities and programs performance capacity of installed linecards.

This information identifies the supervisor, fabric, and linecard modules in the modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

**Command Mode**

Privileged EXEC

**Command Syntax**

```
show platform sand compatibility
```

**Related Commands**

- `platform sand fabric mode (7500 and 7500E Series)` specifies the fabric software mode.
- `platform sand forwarding mode (7500 and 7500E Series)` specifies the forwarding software mode.

**Example**

- This command indicates that the switch is in Fe600 fabric mode and PetraA forwarding mode.

```
switch# show platform sand compatibility

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding mode</td>
<td>None</td>
</tr>
<tr>
<td>Fabric mode</td>
<td>None</td>
</tr>
</tbody>
</table>

PetraA

Fe600

switch#
```
show platform sand lag hardware-only

The `show platform sand lag hardware-only` command displays whether or not LAGs are hardware-only.

**Command Mode**
- Privileged EXEC

**Command Syntax**
- `show platform sand lag hardware-only`

**Examples**
- **This command indicates that LAGs are hardware-only.**
  ```
  switch(config)#platform sand lag hardware-only
  switch(config)#exit
  switch#show platform sand lag hardware-only
  Hardware resources are used for all LAGs: True
  switch#
  ```
- **This command indicates that LAGs are not hardware-only.**
  ```
  switch(config)#no platform sand lag hardware-only
  switch(config)#exit
  switch#show platform sand lag hardware-only
  Hardware resources are used for all LAGs: False
  switch#
  ```
show schedule

The `show schedule` command displays logging output on the terminal during the current terminal session. This command affects only the local monitor. The `no terminal monitor` command disables direct monitor display of logging output for the current terminal session.

The `show schedule` command displays the list of active scheduled commands and the summary of specified scheduled command.

**Command Mode**

Global Configuration

**Command Syntax**

```
show schedule [schedule_name | summary]
```

**Parameters**

- `schedule_name` displays the summary of the specified scheduled command
- `summary` displays the list of active scheduled commands

**Examples**

- This command displays the summary of the “thelp” schedule.

```
switch(config)#show schedule thelp
The last CLI command failed with exit status 1
CLI command "show THelp" is scheduled next at "02:02:35 06/19/2018", interval is 60 minutes
Timeout is 40 minutes
Maximum of 100 log files will be stored
Verbose logging is off
100 log files currently stored in flash:/schedule/thelp

Start time                      Size          Filename
----------------------- ---------------- ----------------------------------
Jun 19 2018 01:02       60.0 bytes       ro301_thelp_2018-06-19.0102.log.gz
Jun 19 2018 00:02       60.0 bytes       ro301_thelp_2018-06-19.0002.log.gz
Jun 18 2018 23:02       60.0 bytes       ro301_thelp_2018-06-18.2302.log.gz
Jun 18 2018 22:02       60.0 bytes       ro301_thelp_2018-06-18.2202.log.gz
Jun 18 2018 21:02       60.0 bytes       ro301_thelp_2018-06-18.2102.log.gz

<--------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command displays the summary of scheduled commands.

```
switch(config)#show schedule summary
Maximum concurrent jobs 1
Prepend host name to logfile: Yes
Name            At time      Last time (mins) Interval (mins) Timeout (mins) Max log files Logfile Location               Status
-------------- ------------- -------------- -------------- -------------- ------------- ------------------------------- ----- 
tech-support    now           00:29          60             30             100                  flash:schedule/tech-support/ Success
thelp           12:02:00     00:02          60             40             100                  flash:schedule/thelp/ Fail
06/05/2018

switch(config)#
```
show version

The show version command displays information that identifies the switch, including its model number, serial number, and system MAC address. The command also provides hardware and software manufacturing information, along with the available memory and elapsed time from the most recent reload procedure.

Command Mode
EXEC

Command Syntax
show version [INFO_LEVEL]

Parameters
- INFO_LEVEL  Specifies information the command displays. Options include
  - <no parameter>  Model and serial numbers, manufacturing data, uptime, and memory.
  - detail  Data listed <no parameter> option plus version numbers of internal components.

Related Commands
- show module displays model and serial numbers of modular system components.

Examples
- This command displays the switch’s model number, serial number, hardware and software manufacturing information, uptime, and memory capacity,

  switch>show version
  Arista DCS-7150S-64-CL-F
  Hardware version: 01.01
  Serial number: JPE13120819
  System MAC address: 001c.7326.fd0c

  Software image version: 4.13.2F
  Architecture: i386
  Internal build version: 4.13.2F-1649184.4132F.2
  Internal build ID: eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

  Uptime: 1 hour and 36 minutes
  Total memory: 4017088 kB
  Free memory: 1473280 kB

  switch>
terminal length

The *terminal length* command overrides automatic pagination and sets pagination length for all show commands on a terminal. If the output of a show command is longer than the configured terminal length, the output will be paused after each screenful of output, prompting the user to continue.

To disable pagination for an SSH session, set *terminal length* to 0. By default, all console sessions have pagination disabled.

The *no terminal length* and *default terminal length* commands restore automatic pagination by removing the *terminal length* command from *running-config*.

The pagination setting is persistent if configured from Global Configuration mode. If configured from EXEC mode, the setting applies only to the current CLI session. Pagination settings may also be overridden when you adjust the size of the SSH terminal window, but can be reconfigured by running the *terminal length* command again.

**Command Mode**

EXEC

**Command Syntax**

```
terminal length lines
no terminal length
default terminal length
```

**Parameters**

- *lines* number of lines to be displayed at a time. Values range from 0 through 32767. A value of 0 disables pagination.

**Example**

- This command sets the pagination length for the current terminal session to 10 lines.

  switch#`terminal length 10`

  Pagination set to 10 lines.

- This command configures the switch to paginate terminal output automatically based on screen size for the current terminal session.

  switch#`no terminal length`

- These commands disable pagination globally.

  switch#`configure`
  switch(config)#`terminal length 0`

  Pagination disabled.
terminal monitor

The terminal monitor command enables the display of logging output on the terminal during the current terminal session. This command affects only the local monitor. The no terminal monitor command disables direct monitor display of logging output for the current terminal session.

Command Mode
Privileged EXEC

Command Syntax
terminal monitor
no terminal monitor
default terminal monitor

Example
• This command enables the display of logging to the local monitor during the current terminal session.
  switch#terminal monitor
  switch#
**trigger**

The `trigger` command specifies what event will trigger the event handler. Handlers can be triggered either by the system booting or by a change in a specified interface’s IP address or operational status.

To specify the action to be taken when the handler is triggered, use the `action bash` command.

**Command Mode**  
Event-Handler Configuration

**Command Syntax**  
`trigger EVENT`

**Parameters**

- **EVENT** event which will trigger the configuration mode event handler. Values include:
  - `onboot` triggers when the system reboots, or when you exit event-handler configuration mode. This option takes no further arguments, and passes no environment variables to the action triggered.
  - `on-intf INTERFACE CHANGE` triggers when a change is made to the specified interface.
  - `on-startup-config` triggers when a change is made to the `startup-config` file.
  - `vm-tracer vm` triggers when a virtual machine monitored by VM Tracer changes state.

- **INTERFACE** the triggering interface. Values include:
  - `ethernet number` Ethernet interface specified by `number`.
  - `loopback number` loopback interface specified by `number`.
  - `management number` management interface specified by `number`.
  - `port-channel number` channel group interface specified by `number`.
  - `vlan numver` VLAN interface specified by `number`.

- **CHANGE** the change being watched for in the triggering interface. Values include:
  - `ip` triggers when the IPv4 address of the specified interface is changed.
  - `ip6` triggers when the IPv6 address of the specified interface is changed.
  - `operstatus` triggers when the operational status of the specified interface changes.

**Examples**

- This command configures the event handler “Eth5” to be triggered when there is a change in the operational status or IP address of Ethernet interface 5.

  ```bash
  switch(config-handler-Eth5)#trigger on-intf Ethernet 5 operstatus ip
  switch(config-handler-Eth5)#
  ```

- This command configures the event handler “onStartup” to be triggered when the system boots, or on exiting event-handler configuration mode.

  ```bash
  switch(config-handler-onStartup)#trigger onboot
  switch(config-handler-onStartup)#
  ```