VITESSE®

ICLI Configuration Guide

Configuration Guide

AN-1104
Revision 1.1
October 2013

Vitesse
Corporate Headquarters
741 Calle Plano
Camarillo, California 93012
United States

www.vitesse.com

TERMS OF USE

The information provided by Vitesse Semiconductor Corporation ("Vitesse") in this document pursuant to these terms ("Agreement") is intended for illustrative purposes only. All information provided herein is subject to change at any time without notice. The information provided, including but not limited to Sample Code ("Software"), is protected by United States and other applicable copyright laws and international treaties. Vitesse does not grant You any license, explicitly or implicitly, under any trademark, patent, copyright, mask work protection right, trade secret or any other intellectual property right.

ALL INFORMATION, INCLUDING BUT NOT LIMITED TO THE SAMPLE CODE ("CODE ") SUPPLIED, IS PROVIDED STRICTLY "AS-IS" WITH NO WARRANTIES OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY, MADE WITH RESPECT TO THE INFORMATION TO INCLUDE THE CODE AND ALL ACCOMPANYING WRITTEN MATERIALS, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. YOU ASSUME THE ENTIRE RISK AS TO THE QUALITY, ACCURACY, AND PERFORMANCE OF THE INFORMATION, AND YOU ASSUME ANY AND ALL RISK AND LIABILITY FOR ANY ACTIONS TAKEN BY YOU ON THE BASIS OF ITS ANALYSIS OR OTHER USE OF THE INFORMATION, INCLUDING BUT NOT LIMITED TO MODIFICATIONS TO YOUR PRODUCTS IN LIGHT OF SUCH USE OF INFORMATION, AND YOU HEREBY ACKNOWLEDGE THAT VITESSE SHALL HAVE NO RESPONSIBILITY OR LIABILITY AS A RESULT OF YOUR USE OF INFORMATION PROVIDED HEREUNDER.

IN NO EVENT SHALL VITESSE BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF USE OR INABILITY TO USE THE CODE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This Agreement is governed by the laws of the State of California, without regard to principles of conflicts of laws. Each provision of this Agreement is severable. If a provision is found to be unenforceable, this finding does not affect the enforceability of the remaining provisions of this Agreement. This Agreement is binding on successors and assigns. By accessing the information contained in or referenced by this document, You acknowledge that You have read this Agreement, that You understand it, that You agree to be bound by its terms, and that this is the complete and exclusive statement of the Agreement between You and Vitesse regarding the information and Code.

Contents

Examples

Figures

Tables

Revision History

Revision	Date	Description
Rev 1.1	October 2013	Minor edits
Rev 1.0	October 2013	First release

1 Introduction

This document describes basic usage and configuration of the Industrial Command Line Interface (ICLI) on the device.

The ICLI is a fully comprehensive management interface on the device. It is the only management interface accessible on the serial console, i.e. even if there is no network connectivity the device can still be managed using a serial connection.

Terminal input/output is indicated like in the following. User input is written in **bold**:

```
# show version

MEMORY           : Total=86382 KBytes, Free=70497 KBytes, Max=70496 KBytes
FLASH            : 0x40000000-0x40ffffff, 64 x 0x40000 blocks
MAC Address      : 00-01-c1-00-ad-80
Previous Restart : Cold
...
```

2 Quick Start

The objective of this Quick Start is to:

- Log in and reset configuration to factory defaults
- Set device hostname and admin user password
- Set VLAN 1 IP address
- Verify connectivity using 'ping'
- Display the current configuration and save it to FLASH storage

The following assumes the device is powered on and has a functional connection to a computer using the serial console port on the device (115200 baud, No parity, 8 data bits, 1 stop bit, no flow control).

The computer must be running a terminal emulator such as TeraTerm or PuTTY on Windows, or Minicom on Linux.

2.1 Log In and Reset Configuration to Factory Defaults

Press Enter one or more times until the 'Username:' prompt appears. Then type 'admin' + Enter and at the 'Password:' prompt press Enter (there is no default password). This completes the login sequence and displays the prompt, '#'.

```
Username: admin
Password:
#
```

At this point the 'admin' user is operating at the highest *privilege level*, level 15. This means full control over the device and its configuration, and it is therefore possible to reset the configuration to factory defaults:

```
# reload defaults
% Reloading defaults. Please stand by.
#
```

When the prompt returns, the system has reverted to factory defaults.

2.2 Set Device Hostname and admin User Password

The ICLI has several different *modes*. The current mode is called *exec* mode; it allows the user to perform operations related to configuration files, reloading defaults, displaying system information, etc., but it does not allow the user to change detailed configuration. Such operations are performed while in the *configuration mode*.

Thus, in order to set the device hostname, first change to configuration mode, then enter the 'hostname' command along with a suitable name, and finally 'exit' configuration mode:

```
# configure terminal
(config)# hostname my-device
my-device(config)# exit
my-device#
```

The commands are executed immediately, so 'hostname' changes the device hostname right away; this is reflected in the prompt as well.

A password should be set for the 'admin' user:

```
my-device# configure terminal
my-device(config)# username admin privilege 15 password unencrypted very-secret
my-device(config)# exit
my-device#
```

The user 'admin' now has password 'very-secret'. Other users can be added in similar fashion, for more details see section 3.5.

2.3 Set VLAN 1 IP Address

The objective is to assign an IP address to the device on VLAN 1. This is often sufficient for small local area networks that use Dynamic Host Configuration Protocol, DHCP[1], or static IP address allocation.

The configuration proceeds in the same manner as setting the hostname: Enter configuration mode, input and execute configuration commands, leave configuration mode:

```
my-device# configure terminal
my-device(config)# interface vlan 1
my-device(config-if-vlan)# ip address dhcp fallback 172.16.1.2 255.255.0.0
my-device(config-if-vlan)# exit
my-device(config)#
```

Notice how the prompt changes; the 'interface vlan 1' command enters a configuration *sub-mode* that allows, among other things, configuration of IP address.

Also note that IP addresses can only be assigned to *VLAN interfaces*.

After configuration is complete the resulting IP address can be inspected; in the below the DHCP negotiation succeeded and the device obtained an address:

```
my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.17/16       DHCP     UP
my-device#
```

'show ip interface brief' displays configured and active IP interfaces. Note how the status should be 'UP'. If it isn't then the reason could be that there is no link on any port.

[1] The system implements a DHCP client that, once enabled, will send out requests for IP address configuration. Those requests are received by a DHCP server on the network (if present and appropriately configured). The server will then search through its pool of available IP addresses, allocate one and return it to the DHCP client. The returned information typically includes IP address, netmask and default gateway, but may also contain other information such as Domain Name Service server addresses.

If DHCP negotiation failed then the fallback IP of 172.16.1.2 would be assigned.

Now the most basic system configuration is complete. Management connectivity can be verified by issuing a 'ping' command to a well-known external IP address:

```
my-device# ping ip 172.16.1.1
PING server 172.16.1.1, 56 bytes of data.
64 bytes from 172.16.1.1: icmp_seq=0, time=0ms
64 bytes from 172.16.1.1: icmp_seq=1, time=0ms
64 bytes from 172.16.1.1: icmp_seq=2, time=0ms
64 bytes from 172.16.1.1: icmp_seq=3, time=0ms
64 bytes from 172.16.1.1: icmp_seq=4, time=0ms
Sent 5 packets, received 5 OK, 0 bad
my-device#
```

If the ping is successful, network logins can now be performed via *telnet* or *ssh* to the address on VLAN interface 1, 172.16.1.17 (or 172.16.1.2).

2.4 Display and Save Configuration to FLASH

The current configuration of the device can be displayed in the form of a virtual file containing the full set of commands necessary to create an identical configuration[2]. This file is called 'running-config' and is volatile by nature; it does not survive across reboots. It is therefore necessary to save 'running-config' to FLASH storage under the name 'startup-config' – this file is read and executed upon every boot and is therefore responsible for restoring the running configuration of the system to the state it had when the saving took place.

'running-config' can be displayed with this command (some details edited out for brevity. Also, the set of interfaces is dependent on hardware capabilities):

```
my-device# show running-config
Building configuration...
hostname my-device
username admin privilege 15 password encrypted dmVyeS1zZWNyZXQ=
!
vlan 1
 name default
!
spanning-tree mst name 00-01-c1-00-ad-80 revision 0
! [...]
!
interface GigabitEthernet 1/1
!
interface GigabitEthernet 1/2
!
! [...]
!
interface 2.5GigabitEthernet 1/1
!
interface 2.5GigabitEthernet 1/2
!
interface vlan 1
 ip address dhcp fallback 172.16.1.2 255.255.0.0
!
line console 0
!
```

[2] With a few exceptions. Certain items, such as private SSH keys, are not displayed.

```
line vty 0
!
! [...]
!
end
my-device#
```

Lines that begin with '!' are comments. The file begins with the 'hostname' command and the password for user 'admin', then followed by VLANs 1 and 42. Other configuration may display, such as Spanning Tree Protocol (STP). Then follows a list of all port interfaces on the device, ordered by switch ID, type and port number. All interfaces except GigabitEthernet 1/1 are at default settings, so nothing is displayed for them. This is a general rule of thumb: Only non-default configuration is displayed; otherwise the output would be huge and readability would suffer. There are exceptions, though, to be discussed later.

After the physical interfaces follows VLAN interfaces, 1 and 42. Only the latter has an IP address assigned.

Finally follows the 'line' section; it specifies characteristics for the serial console ('line console 0') or network ICLI management connections ('line vty *x*').

The configuration as displayed above is also what is saved to 'startup-config':

```
my-device# copy running-config startup-config
Building configuration...
% Saving 1326 bytes to flash:startup-config

my-device# dir
Directory of flash:
    r- 1970-01-01 00:00:00      648 default-config
    rw 1970-01-03 18:21:28     1326 startup-config
2 files, 1974 bytes total.

my-device# more flash:startup-config
hostname my-device
username admin privilege 15 password encrypted dmVyeS1zZWNyZXQ=
!
vlan 1
 name default
[...]
```

The 'dir' command lists the files in the FLASH file system; 'more' outputs the contents of one of them.

2.5 Summing Up

This concludes the Quick Start. The skills exercised here form the basis for all day-to-day work with the Command Line Interface on the device: Logging in, displaying information ('show'), working with the configuration files ('show running-config', 'copy', 'dir', 'more'), working with the actual configuration ('configure terminal', 'exit') and sub-modes ('interface ...').

3 ICLI Basics

The ICLI has some key characteristics:

- It is *modal*, i.e. certain operations are possible or impossible in specific modes
- It is *line-based*, i.e. no screen editing features
- It executes commands instantly upon end-of-line
- It is *privilege-based*, i.e. certain operations require the user to have a certain *privilege level* to succeed
- It implements industrial de-facto behavior for network equipment CLIs, i.e. it structurally and behaviorally resembles CLIs found on other equipment while still possessing unique characteristics in some areas.

The ICLI can be accessed directly via the serial console, or over the network via *telnet* or *ssh*. In each case the user has to log in before ICLI commands can be executed; this begins a *session* which lasts until log out.

Multiple sessions can co-exist at the same time, each providing separate environments: Logged-in user ID, privilege level, command history, mode and session settings. It is therefore perfectly possible for the same user to control several concurrent sessions; for example, one serial console session and one ssh session.

The user database is either local or provided by a RADIUS or TACACS+ server. In case of a local user database, passwords and privilege levels are maintained on the device.

3.1 Command Structure and Syntax

A *command* is a single line of text consisting of *keywords* and *parameters*, for example:

```
my-device# show vlan id 10
...
my-device# show vlan id 20
...
```

The keywords are ‘show’, ‘vlan’, and ‘id’; whereas 10 and 20 are parameters, something that could contain another value in another command invocation.

Keywords are case-insensitive, i.e. ‘show’, ‘SHOW’ and ‘Show’ are identical. Conversely, parameters may either be case-sensitive or case-insensitive depending on the command and parameter in question.

Keywords and certain parameters can be abbreviated as long as they are unambiguous. For example, these commands are identical:

```
my-device# show interface GigabitEthernet 1/5 capabilities
...
my-device# sh in g 1/5 c
...
```

This works because:

1. There are many keywords that begin with 's' but only one that begins with 'sh'
2. There are several commands that begin with 'show i' but only one that begins with 'show in'
3. The 'show interface' command takes a port type as parameter. Depending on the hardware capabilities, the options are: FastEthernet, GigabitEthernet, 2.5GigabitEthernet, 5GigabitEthernet and 10GigabitEthernet. Thus, 'g' is a unique abbreviation for 'GigabitEthernet'
4. 1/5 identifies the interface as belonging to switch 1, port 5. This parameter cannot be abbreviated and has to be written out in full
5. The 'show interface GigabitEthernet 1/5' command can output different kinds of information: Capabilities, statistics, status, and several other. In this case 'c' is a unique abbreviation for 'capabilities'

With a bit of practice this allows for highly efficient keyboard entry, in particular when coupled with the context-sensitive help features of the ICLI (see section 3.3.3, Context-sensitive Help).

3.1.1 Syntax

A command is described by its *syntax*[3], for example:

show interface *list* { **status** | **statistics** | **capabilities** | **switchport** | **veriphy** }

and

show erps [*groups*] [**detail** | **statistics**]

The semantics are:

- **keyword**s are written in bold
- *parameters* are written in italics
- [...] indicates an optional construct: It may or may not be present
- { ... } indicates a grouping; the constructs within belong together somehow
- '|' indicates a choice between two or more alternatives, e.g. **a** | **b** | **c** which reads as "a or b or c".

Thus, the first command syntax is simple: First 'show', then 'interface', then a list of interfaces, then exactly one of 'status', 'statistics', 'capabilities', 'switchport' and 'veriphy'.

The second command is a bit more complex: 'show' and 'erps' are mandatory, but then the remaining parameters and keywords are optional: The user *may* enter group IDs; the user *may* enter either 'statistics' or 'detail', e.g.:

[3] The syntax is represented in slightly different manner in this documentation and in a CLI session. In this document, variable parameters are written in *italics*, whereas a CLI session will display such items surrounded by '<' and '>'.

```
! Show short-form ERPS (Ethernet Ring Protection Switching) information for all
! instances:
my-device# show erps
...

! Show statistics for all instances:
my-device# show erps statistics
...

! Show details for all instances:
my-device# show erps detail
...

! But it is not allowed to show details and statistics at the same time:
my-device# show erps detail statistics
                           ^
% Invalid word detected at '^' marker.

! Show details for specific set of instances:
my-device# show erps 1-6 detail
...
```

There are some slightly more complex features of the syntax that center around sequences of optionals such as [**a**] [**b**] [**c**].

- Each of a, b, c may or may not be present, e.g. "a c" is valid, as is no input at all
- Order is not important, e.g. "a c" and "c a" are equivalent
- Each optional can be present exactly zero or one time, i.e. not repeated

There are variations:

- Group of optionals of which at least one must be present: { [**a**] [**b**] [**c**] }*1
- Group of optional where one or more has fixed position: [**a**] {[**b**]} [**c**]
    - This says that 'b' is optional, but *if* it is present *then* it must follow after 'a' (if 'a' is present) *and* it must come before 'c' (if 'c' is present)

For example, assume a command with this syntax:

```
a [b] [c] { d | e } {[f] [g]}*1
```

then valid input examples are:

- 'a d f', because 'b' and 'c' are optional, 'd' is picked instead of 'e', and 'f' is chosen as the mandatory optional
- 'a d f g', because 'b' and 'c' are optional, 'd' is picked instead of 'e', and both 'f' and 'g' are chosen in the final group of optional
- 'a c b e g', because the 'b' optional is omitted, 'e' is picked instead of 'd', and 'g' is chosen for the mandatory optional

3.2 Ethernet Interface Naming

An Ethernet interface ("port") is identified by three pieces of information:

- Its type: FastEthernet, GigabitEthernet, 2.5GigabitEthernet, 5GigabitEthernet, 10GigabitEthernet
- The switch it belongs to. For non-stacking systems this value is always 1
- The port number within the type and switch; the numbering starts with 1 for each type, so a switch may have e.g. both GigabitEthernet 1/1 and 2.5GigabitEthernet 1/1

Many ICLI commands accept a list of interfaces. In its simplest form such a list is a sequence of (type, switch ID, port) information separated by whitespace, e.g.: 'GigabitEthernet 1/3 10GigabitEthernet 1/5'.

As can be seen this allows a single list to mix different types.

The switch ID and the port numbers can be listed either as single numbers, as lists or as sequences. A list is a comma-separated set of single port numbers or sequences, whereas a sequence is of the form: *from—to*.

Some examples:

- GigabitEthernet 1/5 for the single gigabit port number 5 on switch 1
- GigabitEthernet 1/2,4,10-12 for gigabit ports 2, 4, 10, 11, 12 on switch 1
- GigabitEthernet 1-3/2 for gigabit port 2 on switches 1, 2 and 3

It is possible to *wildcard* the type and/or switch ID and/or ports to mean "all types", "all switch IDs" and "all ports", respectively. A wildcard is written with an asterisk instead of type, switch ID or port, and some further abbreviations are possible:

- '*' means "all ports of all types on all switches"
- *type* '*' means "all ports of the specified type on all switches"

Some examples to clarify. Assume a stack with two switches, switch ID 1 and 3. Each switch has 9 gigabit ports and two 2.5 gigabit ports. Then:

- interface * (or: interface * * *)
    - All ports of all types on all switches: GigabitEthernet 1,3/1-9 2.5GigabitEthernet 1,3/1-2
- interface * 1/2
    - Switch 1, port number 2 of all types: GigabitEthernet 1/2 2.5GigabitEthernet 1/2
- interface * */2
    - All switches, all types, port number 2: GigabitEthernet 1,3/2 2.5GigabitEthernet 1,3/2
- interface * */4
    - All switches, all types, port number 4: GigabitEthernet 1,3/4

      (Note how there are no 2.5 gigabit ports in the result)
- interface GigabitEthernet 3/*

    - Switch 3, all gigabit ports: GigabitEthernet 3/1-9
- interface 2.5GigabitEthernet * (or: interface 2.5GigabitEthernet */*)
    - All 2.5 gigabit ports on all switches: 2.5GigabitEthernet 1,3/1-2

Wildcards will include the largest possible set of ports, but may output an error message if a specific switch ID or port number doesn't exist.

For example, these sets are invalid:

- interface * 2/*
    - All ports of all types on switch 2 – which isn't a member of the stack
- interface * */100
    - There is no port 100 of any type on any switch
- interface GigabitEthernet */* 2.5GigabitEthernet 2/*
    - Again, switch 2 doesn't exist so the entire set is considered invalid

Validity is determined per set of (type, switch ID, port) containing wildcards: The result for that set is valid if there is at least one port that matches the set. A list of sets is valid if all sets match at least one port each.

3.3 Using the Keyboard

The ICLI provides a rich set of keys to assist the user while working with the command line. The functionality is divided into:

- Basic line editing
- Command history
- Context-sensitive help
- Long lines and pagination

3.3.1 Basic Line Editing

Basic line editing allows the input of characters to form a command line, while also allowing cursor movement and insertion/deletion of characters and words. The available editing functions and keys are:

Table 1. Basic Line Editing Keys

Key	Operation
Left / Right	Move one character left/right
Home / Ctrl-A	Move to start of line
End / Ctrl-E	Move to end of line
Del / Ctrl-D	Delete character at cursor

Key	Operation
Backspace / Ctrl-H	Delete character to the left of cursor
Ctrl-N	Delete the entire current line
Ctrl-U / Ctrl-X	Delete all characters to the left of the cursor
Ctrl-K	Delete all characters under the cursor and right
Ctrl-W	Delete from cursor to start of word on the left
TAB	Complete word at end-of-line (see also section 3.3.3)

3.3.2 Command History

A session maintains a non-persistent command history of previously entered command lines. The history can be up to 32 lines long; once full, a new line will push the oldest entry out.

Table 2. Command History Keys

Key	Operation
Up / Ctrl-P	Previous line in command history
Down	Next line in command history

The number of lines to keep in the history for the current session is configurable:

```
my-device# terminal history size 32
```

The size is a value between 0 and 32; 0 disables the history entirely.

The current value is displayed as part of the output from 'show terminal':

```
my-device# show terminal
Line is con 0.
    * You are at this line now.
    Alive from Console.
    Default privileged level is 2.
    Command line editing is enabled
    Display EXEC banner is enabled.
    Display Day banner is enabled.
    Terminal width is 80.
             length is 24.
             history size is 32.
             exec-timeout is 10 min 0 second.

    Current session privilege is 15.
    Elapsed time is 0 day 0 hour 6 min 20 sec.
    Idle time is 0 day 0 hour 0 min 0 sec.
```

It is possible to list the history:

```
my-device# show history
  show running-config
  copy running-config startup-config
  dir
  show history
my-device#
```

The list begins with the oldest entry at top ('show running-config').

3.3.3 Context-sensitive Help

The ICLI implements several hundred commands ranging from the very simple to the very complex. It is therefore imperative that the user can be assisted in entering syntactically correct commands as well as discovering relevant commands. These objectives are supported by the context sensitive help features.

Table 3. Context-sensitive Help Keys

Key	Operation
?	Show next possible input and description
? ? / Ctrl-Q	Show syntax of possible command(s)
TAB	Show next possible input without description or expand current word fully if it is unambiguous

The context-sensitive help only displays commands that are accessible at the current session privilege level (see section 3.6).

Example 1. Using Context-sensitive Help

```
! Show possible next input for a command that begins with 'show a':
my-device# show a?
    aaa            Login methods
    access         Access management
    access-list    Access list
    aggregation    Aggregation port configuration

! The same, but without descriptions:
my-device# show a<TAB>
aaa          access       access-list  aggregation

! If the user enters another 'g' the word 'aggregation' is the only possibility:
my-device# show ag?
    aggregation    Aggregation port configuration
    <cr>

! Pressing <TAB> now expands the word fully:
my-device# show aggregation

! Possible next input is displayed with a press of '?':
my-device# show aggregation ?
    |       Output modifiers
    mode    Traffic distribution mode
    <cr>

! The syntax is displayed with another press of '?':
my-device# show aggregation ?
show aggregation [ mode ]

! This shows that there is an optional 'mode' word (square brackets indicate an
option).

! Repeated presses of '?' toggles display between next possible input and syntax:
my-device# show aggregation ?
    |       Output modifiers
    mode    Traffic distribution mode
    <cr>
```

```
my-device# show aggregation ?
show aggregation [ mode ]

! Finally, the syntax display is also directly available with Ctrl-Q:
my-device# show aggregation ^Q
show aggregation [ mode ]
```

3.3.4 Long Lines and Pagination

A session has configuration that indicates the width of the terminal in characters and the height in lines. It uses these parameters to control handling of long input lines and to control pagination of multi-line output. For details about changing these parameters please refer to section 3.7.

Long lines come into play when a line is longer than the terminal width minus the prompt. In that case part of the line will be hidden from display, as indicated by '$' at the beginning and/or end of the visible part of the line.

For example:

```
my-device# $there is text to the left of what is visible here
my-device# there is text to the right of what is visible here$
my-device# $there is text at both ends of what is visible here$
```

The first line has scrolled left; the second line has scrolled right; the third line has been scrolled to the middle of a quite long line.

Pagination appears each time execution of a command causes output of more lines than what has been configured as terminal length. A typical example is the output from 'show running-config'. After the first several lines have been output, the pagination prompt is presented:

```
! [lines of text]
-- more --, next page: Space, continue: g, quit: ^C
```

The following keys control pagination:

Table 4. Pagination Control Keys

Key	Operation
Enter	Display next line of output
Space	Display next page of output
G	Display remainder of output without more pagination
Q / Ctrl-C	Discard remainder of output
Any other key	Display next page of output. Note that certain terminal keys (arrows, Home, End, etc.) may appear as multiple characters to the ICLI, leading to multiple pages being output in quick succession.

The terminal length (also sometimes called height) can be configured for the current session using the 'terminal length *lines*' command. If *lines* = 0 is input, pagination is disabled.

```
my-device# terminal length 0
my-device# terminal length 25
```

3.3.5 Other Special Keys

One additional key is defined as a convenience. It allows the immediate return from any sub-mode to exec mode (see the following section).

Table 5. Other Special Keys

Key	Operation
Ctrl-Z	Return directly to Exec mode

3.4 Filtering Output

The output from commands can in most cases be *filtered*: It is possible to limit the output to only those lines that match/trigger a specific substring. The available filtering is:

- Begin – display the first line that matches and all subsequent lines
- Include – display exactly those lines that match
- Exclude – display exactly those lines that do not match

The string is case-sensitive.

The syntax is:

command '|' { **begin** | **include** | **exclude** } *string*

```
! Execute a command that generates some output; no filtering initially:
my-device# show users
Line is con 0.
    * You are at this line now.
    Connection is from Console.
    User name is admin.
    Privilege is 15.
    Elapsed time is 0 day 21 hour 52 min 50 sec.
    Idle time is 0 day 0 hour 0 min 0 sec.

! Filter to include specific word:
my-device# show users | include User
    User name is admin.

! Exclude all lines that contain '0' (zero)
my-device# show users | exclude 0
    * You are at this line now.
    Connection is from Console.
    User name is admin.
    Privilege is 15.

! Begin output when specific word is matched:
my-device# show users | begin Elapsed
    Elapsed time is 0 day 21 hour 53 min 29 sec.
    Idle time is 0 day 0 hour 0 min 0 sec.
```

3.5 Understanding Modes and Sub-modes

The ICLI implements a number of *modes* that control the available command set. The modes are further influenced by the *privilege level* of the user; some modes or commands are only accessible to administrators while others require no privileges beyond log in.

There are three major modes, Exec, Privileged Exec and Config; and under Config there exist a number of *sub-modes*. The sub-modes allow configuration of specific VLANs, Ethernet interfaces, etc.

Table 6. Modes and Sub-modes

Mode	Parent Mode	Description
Exec	-	Lowest-privileged mode; used for basic system monitoring. Generally does not allow modifications to the system. Command: `disable` Prompt: *hostname*>
Privileged Exec	Exec	Privileged mode; allows configuration and other modifications to the system. Command: `enable` Prompt: *hostname*#
Config	Priv. exec	Global configuration mode Command: `configure terminal` Prompt: *hostname*(config)#
VLAN config	Config	Sub-mode for configuring active VLANs Command: `vlan` *vlan_id_list* Prompt: *hostname*(config-vlan)#
VLAN interface config	Config	Sub-mode for configuring VLAN interfaces Command: `interface vlan` *vlan_id_list* Prompt: *hostname*(config-if-vlan)#
Interface config	Config	Sub-mode for configuring Ethernet interfaces Command: `interface` *type switch_num/port_num* Prompt: *hostname*(config-if)#
Line	Config	Sub-mode for configuring terminal lines Command: `line { con | vty }` *line_num* Prompt: *hostname*(config-line)#
IPMC Profile Config	Config	Sub-mode for configuring IP Multicast profiles Command: `ipmc profile` *profile_name* Prompt: *hostname*(config-ipmc-profile)#
SNMP Server Host Config	Config	Sub-mode for configuring SNMP server host entries Command: `snmp-server host` *host_name* Prompt: *hostname*(config-snmps-host)#

Mode	Parent Mode	Description
STP Aggregation Config	Config	Sub-mode for configuring Spanning Tree Protocol aggregation Command: `spanning-tree aggregation` Prompt: *hostname*`(config-stp-aggr)#`
DHCP Pool Config	Config	Sub-mode for configuring DHCP client pools Command: `ip dhcp pool` *pool_name* Prompt: *hostname*`(config-dhcp-pool)#`
RFC2544 Profile Config	Config	Sub-mode for configuring RFC2544 profiles Command: `rfc2544 profile` *profile_name* Prompt: *hostname*`(config-rfc2544-profile)#`

It is possible for a user to transition between these modes using certain commands, subject to the user's privilege level and the current session privilege level (see section 3.6).

The initial mode is determined by the privilege level of the user logging in. If the privilege level is zero or one the user is *unprivileged* and begins in the (Unprivileged) Exec mode. If the privilege level is higher the session begins in Privileged Exec mode.

A user can raise the Exec mode privilege level to a higher value if an *enable password* has been configured for that level. This elevation is done with the 'enable *level*' command, where level is a value between 1 and 15. The reverse operation, lowering the privilege level, is achieved with the 'disable' command.

Once in Privileged Exec mode it is possible to enter into the Global Configuration mode by entering the command 'configure terminal'. Exit from Global Configuration is achieved with one of 'end', 'exit' or Ctrl-Z.

Access to a configuration sub-mode (e.g. for Ethernet interfaces) goes through Global Configuration or another sub-mode, i.e. it is possible to change directly from, say, VLAN sub-mode to Ethernet interface sub-mode.

Each mode and sub-mode thus implements a *scope* for commands: Inside each mode a particular subset of commands is available; to get to other commands one must generally change mode/sub-mode. This is necessary because there are commands with identical prefixes in different modes; for example there are commands that begin with 'ip' in Privileged Exec, Global Configuration and VLAN Interface Configuration modes.

There are two exceptions to this:

- While in a configuration sub-mode, access to Global Configuration mode commands is possible as long as there is no ambiguity. Execution of a Global Configuration command exits the sub-mode.
- Exec mode commands, be that privileged or unprivileged, are accessible from within Global Configuration or one of the sub-modes by using the 'do' command.

The 'do' command takes an arbitrary command line from Exec and executes it. In the following example, the user wants to change the IP address on the VLAN 1 interface, but wants to verify the current address while in the sub-mode.

Example 2. Using 'do' While In a Sub-mode

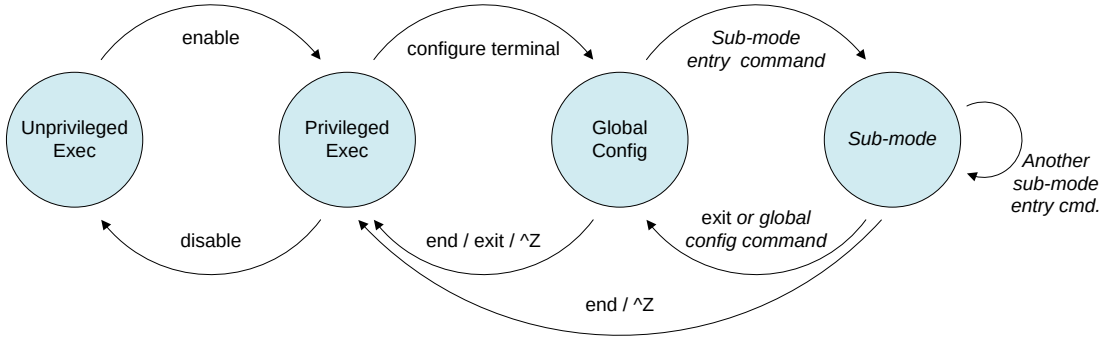
```
my-device# configure terminal
my-device(config)# interface vlan 1
my-device(config-if-vlan)# do show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.15/24       DHCP     UP
my-device(config-if-vlan)# end

! When in Exec, no 'do' prefix is needed:
my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.15/24       DHCP     UP
```

3.5.1 ICLI Mode Transitions

The following figure illustrated the possible transitions between major modes and sub-modes, and some of the relevant commands.

Figure 1. ICLI Mode Transition Commands



Example 3. Changing Between ICLI Modes

```
! Initial mode for this example is Unprivileged Exec. Raise level
! (and change mode):
my-device> enable
Password: ***
my-device#

! Note how the prompt changed from '>' to '#' to indicate the privileged exec mode

! Enter Global Configuration mode:
my-device# configure terminal

! Now create VLAN 100 and give it a name. This enters the VLAN sub-mode, as
! indicated by a new prompt:
my-device(config)# vlan 100
my-device(config-vlan)# name MyVlan

! Change directly from VLAN sub-mode into Ethernet interface sub-mode for
! interface instance 4 on switch 1, and set link speed to 'auto'
my-device(config-vlan)# interface GigabitEthernet 1/4
```

```
my-device(config-if)# speed auto

! Then enter a command from the global configuration mode; this leaves Ethernet
! interface sub-mode
my-device(config-if)# hostname my-device

! Exit Global Configuration mode and go back to Privileged Exec
my-device(config)# end

! And use 'disable' to go back to Unprivileged Exec:
my-device# disable
my-device>
```

3.6 Understanding Privilege Levels

A *privilege level* is a number in the range 0 to 15, inclusive, with 0 being the lowest. It is assigned to a user session and used to determine access to ICLI commands: Only commands at the same or lower privilege level can be accessed.

Each user on the device has a default privilege level which is copied to the session's privilege level at log in. It is, however, possible for the user to change the session privilege level by executing the 'enable' or 'disable' commands. This can be used, for example, as follows:

- The user account is configured with privilege level 0
- Whenever the user needs to perform higher-privileged commands, the user changes session priority level, executes the necessary commands, and then revert back to the default priority level

Access to higher priority levels must be password protected by using the 'enable password' or 'enable secret' global configuration commands. The main difference between the two is whether passwords are displayed in clear text or encrypted form in running-config (and, consequently, startup-config).

Password input can also be in encrypted or clear text form. The latter is used when an operator input a new password, as the operator will usually not know the encrypted form of the password.

The 'admin' user is by default at level 15, i.e. at the highest possible privilege level.

Example 4. Configuring Privilege Level Passwords

The following example configures a level 15 password using 'enable secret', inspects the resulting configuration, then removes it again.

```
my-device# configure terminal

! A secret can either be input in clear text or encrypted form; a digit indicates
! which kind follows on the command line:
my-device(config)# enable secret ?
    0    Specifies an UNENCRYPTED password will follow
    5    Specifies an ENCRYPTED secret will follow

! In this case: Unencrypted. Then follows either the level for which a password
! is being configured, or, if no level is given, the password for level 15:
my-device(config)# enable secret 0 ?
    <word32>    Password
```

```
    level       Set exec level password

! Thus, the following two commands are semantically identical:
my-device(config)# enable secret 0 my-secret
my-device(config)# enable secret 0 level 15 my-secret

! The running configuration can be inspected to see the encrypted form:
my-device(config)# do show running-config | include enable
enable secret 5 level 15 D29441BF847EA2DD5442EA9B1E40D4ED

! To remove the password use the 'no' form (the two are semantically equivalent
for level 15):
my-device(config)# no enable secret
my-device(config)# no enable secret level 15
my-device(config)# do show running-config | include enable
my-device(config)#
```

3.7 Understanding Terminal Parameters

Each login to the system, be that via the serial console or via telnet or ssh, creates a session. The session is initialized with settings that are configurable from the 'line' configuration sub-mode, but most of them can also be changed from exec mode while the session is active. Such changes are not persistent, however, and are lost when the session is terminated.

The below table lists the available settings and the modes where each can be configured.

Table 7. Terminal Parameters

Setting	Modes	Description
editing	Exec, Line	Enable/disable command line scrolling
exec-banner	Line	Enable/disable display of the Exec banner (configured with 'banner exec ...')
exec-timeout	Exec, Line	Inactivity timer; automatically log out after a period of inactivity. A value of zero disables automatic logout
history	Exec, Line	Length of command history buffer
length	Exec, Line	Terminal length in lines, used for pagination. Zero disables pagination
location	Line	A line of text that describes the terminal location, e.g. "Server room"
motd-banner	Line	Enable/disable display of Message-Of-The-Day banner (configured with 'banner motd ...')
privilege	Line	Assign default privilege level
width	Exec, Line	Terminal width in characters, used for pagination

The system allows one serial console session and up to 16 network sessions. The console session is called "console 0" whereas each network session is called "vty *X*" where X is a value between 0 and 15 (vty is an abbreviation for Virtual TTY).

The configuration appears near the bottom of 'running-config' and looks like this:

```
line console 0
 exec-timeout 0
!
line vty 0
!
line vty 1
!
line vty 2
! [...]
```

It is possible to specify different settings for each vty, but this is generally not recommended since there is no way to associate an incoming ssh or telnet connection with a specific vty.

Example 5. Changing Terminal Parameters

The below example shows how to change a couple of values for the current session, and for all future console sessions.

```
! First inspect current settings for this session:
my-device# show terminal
Line is con 0.
    * You are at this line now.
    Alive from Console.
    Default privileged level is 2.
    Command line editing is enabled
    Display EXEC banner is enabled.
    Display Day banner is enabled.
    Terminal width is 80.
             length is 24.
             history size is 32.
             exec-timeout is 10 min 0 second.

    Current session privilege is 15.
    Elapsed time is 0 day 0 hour 15 min 42 sec.
    Idle time is 0 day 0 hour 0 min 0 sec.

! Then set terminal length to zero to disable pagination, and exec-timeout to
! zero to disable automatic logout:
my-device# terminal length 0
my-device# terminal exec-timeout 0
my-device# show terminal
Line is con 0.
    * You are at this line now.
    Alive from Console.
    Default privileged level is 2.
    Command line editing is enabled
    Display EXEC banner is enabled.
    Display Day banner is enabled.
    Terminal width is 80.
             length is 0.
             history size is 32.
             exec-timeout is 0 min 0 second.

    Current session privilege is 15.
    Elapsed time is 0 day 0 hour 16 min 31 sec.
    Idle time is 0 day 0 hour 0 min 0 sec.

! Then we do the same, but for all future console sessions. Note how the commands
! have no 'terminal' prefix ('terminal length' vs. 'length'):
my-device# configure terminal
my-device(config)# line console 0
```

```
my-device(config-line)# exec-timeout 0
my-device(config-line)# length 0
my-device(config-line)# end

! Finally save the configuration to startup-config to make it persistent:
my-device# copy running-config startup-config
Building configuration...
% Saving 1287 bytes to flash:startup-config
my-device#
```

3.7.1 Using Banners

The system provides three different *banners*; text that is output as messages to the user:

- The Message Of The Day banner (MOTD), displayed upon connection to the system, or when a console login attempt has timed out
- The Login banner, displayed before the first "Username:" login prompt
- The Exec banner, displayed upon successful login

All of the above are configured in a similar manner, using the 'banner' command:

banner [**motd**] *banner*
banner exec *banner*
banner login *banner*

The banner text can be either a single line or multiple lines. The first character of the text defines a *delimiter character*; the actual text of the banner then follows and ends at the first appearance of the delimiter character. Neither of the delimiters are included in the actual text.

Example 6. Configuring Banners

```
! First configure the MOTD banner, which in this case is multi-line. '*' is
! used as delimiter character, but any printable character that isn't used in
! the message is usable:
my-device# configure terminal
my-device(config)# banner motd *This is the Message Of The Day Banner.
Enter TEXT message.  End with the character '*'.
It spans multiple lines.
And one more. But now it ends.*

! Then the Login and Exec banners. Both are single-line. Note how different
! delimiters are used in each banner:
my-device(config)# banner login XThis is my-device.X
my-device(config)# banner exec "WARNING: Production system. Be careful."
my-device(config)# end

! Inspect configuration:
my-device# show running-config
Building configuration...
banner motd "This is the Message Of The Day Banner.
It spans multiple lines.
And one more. But now it ends."
banner exec "WARNING: Production system. Be careful."
banner login "This is my-device."
```

```
hostname my-device
! [...]
end

! Test it: Log out, then log in again:
my-device# exit

This is the Message Of The Day Banner.
It spans multiple lines.
And one more. But now it ends.

Press ENTER to get started<ENTER>

This is my-device.

Username: admin
Password:

WARNING: Production system. Be careful.
my-device#

! Finally save the configuration to startup-config to make it persistent:
my-device# copy running-config startup-config
Building configuration...
% Saving 1461 bytes to flash:startup-config
my-device#
```

4 Configuring the System

Changes to system configuration can only be done from the Global Configuration mode and its sub-modes[4]. The outline is:

i. Raise privilege level to 15

ii. Enter Global Configuration mode

iii. Input appropriate configuration commands

    1. Optionally enter sub-modes and input appropriate commands there

iv. Exit Global Configuration mode

v. Verify configuration

vi. Save configuration to FLASH

Example 7. Configuration

In this example the hostname and VLAN 1 IP address is configured, verified and saved.

```
! This example assumes the session is initially unprivileged.

! Step 1: Raise privilege level:
> enable
Password: ***

! Step 2: Enter Global Configuration mode:
# configure terminal

! Step 3: Input configuration commands. The IP address is set from within the
! VLAN interface submode:
(config)# hostname my-device
my-device(config)# interface vlan 1
my-device(config-if-vlan)# ip address dhcp fallback 172.16.1.2 255.255.0.0
my-device(config-if-vlan)# exit

! Step 4: Leave Global Configuration mode and go back to Privileged Exec:
my-device(config)# end

! Step 5: Inspect and verify the configuration (some output omitted for brevity):
my-device# show running-config
Building configuration...
hostname my-device
username admin privilege 15 password encrypted Zm9v
!
vlan 1
 name default
!
interface GigabitEthernet 1/1
!
```

[4] Except when working with configuration files or reloading defaults. This is done in Privileged Exec mode.

```
interface GigabitEthernet 1/2
!
...
interface vlan 1
 ip address dhcp fallback 172.16.1.2 255.255.0.0
!
...
end

! More verification: Display IP interfaces and assigned IP address and status:
my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.15/24       DHCP     UP

! An address was obtained from DHCP, so the fallback wasn't used

! Try to inspect hostname:
my-device# show hostname
               ^
% Invalid word detected at '^' marker.

! No such command exists, but it is possible to extract a single line from
! running-config by using a filter:
my-device# show running-config | include hostname
hostname my-device

! Step 6: Save configuration to FLASH:
my-device# copy running-config startup-config
Building configuration...
% Saving 1272 bytes to flash:startup-config
```

4.1 Resetting or Removing Configuration Using 'no' Forms

It is possible to either remove specific configuration or reset it to its default values.

In the general case, almost each configuration command has a corresponding 'no' form. The 'no' form is syntactically similar (but not necessarily identical) to the configuration command, but either resets the parameters to defaults for the configurable item being addressed, or removes the item altogether.

In many cases 'no' can be read as "no(t) different from default settings".

Example 8. Using 'no' Forms

The following example:

- Configures the VLAN 1 interface IP address to use DHCP
- Configures the DNS name server to be taken from DHCP
- Inspects the configuration
- Removes the DNS name server
- Removes the IP address on the VLAN 1 interface

Both 'no' operations can be viewed as reset-to-default, with the defaults being: No DNS name server and IP address.

```
my-device# configure terminal
my-device(config)# interface vlan 1
my-device(config-if-vlan)# ip address dhcp
my-device(config-if-vlan)# exit
my-device(config)# ip name-server dhcp
my-device(config)# end

my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.15/24       DHCP     UP

my-device# show ip name-server

Current DNS server is 172.16.1.1 set by DHCP.

my-device# configure terminal
my-device(config)# no ip name-server
my-device(config)# interface vlan 1
my-device(config-if-vlan)# no ip address
my-device(config-if-vlan)# end
my-device# show ip name-server

Current DNS server is not set.
my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
my-device#
```

Note how the syntax of the configuration commands and their 'no' forms are different; the 'no' forms usually do not take as many parameters.

5 Adding/Modifying/Deleting Users

The following describes local user management on the device. RADIUS and TACACS+ user management is beyond the scope of this document.

It is possible to create several user accounts on a system. Each user account has a set of configurable attributes:

- User name
- Password
- Privilege level

All attributes are configured with the same command, 'username'.

username *username* **privilege** *level* **password** { **unencrypted** | **encrypted** } *password*
username *username* **privilege** *level* **password none**
no username *username*

'password none' is used when no password is desired; the security implications of using this should be considered carefully.

'no username' deletes the given user account.

Example 9. Adding, Modifying and Deleting Users

The following example adds two user accounts at different privilege levels, inspects configuration, and deletes one account again using 'no username'.

```
! Display current set of local user accounts:
my-device# show running-config | include username
username admin privilege 15 password encrypted dmVyeS1zZWNyZXQ=

! Add two accounts, 'operator' and 'monitor'. The passwords are supplied in
! unencrypted form:
my-device# configure terminal
my-device(config)# username operator privilege 10 password unencrypted a-secret
my-device(config)# username monitor privilege 1 password unencrypted new-secret

! Verify that the configuration is correct. Note that passwords are displayed
! in encrypted form:
my-device(config)# do show running-config | include username
username admin privilege 15 password encrypted dmVyeS1zZWNyZXQ=
username operator privilege 10 password encrypted YS1zZWNyZXQ=
username monitor privilege 1 password encrypted YW5vdGhlci1zZWNyZXQ=

! Delete the 'operator' user and verify it is removed from the configuration:
my-device(config)# no username operator
my-device(config)# do show running-config | include username
username admin privilege 15 password encrypted dmVyeS1zZWNyZXQ=
username monitor privilege 1 password encrypted YW5vdGhlci1zZWNyZXQ=
```

6 Using ‘show’ Commands

The family of ‘show’ commands is the cornerstone of ICLI-based system monitoring. Most features implement one or more ‘show’ commands that will display a relevant mix of status and configuration.

> Please note: The exact set of available commands, parameters and output format depends on the system configuration and software version, so some of the following commands and examples may not be applicable to all systems.

‘show’ commands exist only in the two Exec modes and are subject to session privilege level enforcement. Therefore, listing the largest possible set of ‘show’ commands requires the session to be at level 15.

Example 10. Listing All ‘show’ Commands

The following example raises the session privilege level to 15. In this example an ‘enable secret’ has been specified, so password entry is required to proceed.

Then the user inputs ‘show’ and uses the context-sensitive help feature to list the possible show commands, in this case for a Carrier Ethernet system.

```
my-device> enable
Password: ***
my-device# show ?
    aaa                 Login methods
    access              Access management
    access-list         Access list
    aggregation         Aggregation port configuration
    clock               Configure time-of-day clock
    dot1x               IEEE Standard for port-based Network Access Control
    eps                 Ethernet Protection Switching
    erps                Ethernet Ring Protection Switching
    evc                 Ethernet Virtual Connections
    green-ethernet      Green ethernet (Power reduction)
    history             Display the session command history
    interface           Interface status and configuration
    ip                  Internet Protocol
    ipmc                IPv4/IPv6 multicast configuration
    ipv6                IPv6 configuration commands
    lacp                LACP configuration/status
    line                TTY line information
    link-oam            Link OAM configuration
    lldp                Display LLDP neighbors information.
    logging             Syslog
    loop-protect        Loop protection configuration
    mac                 Mac Address Table information
    mep                 Maintenance Entity Point
    mvr                 Multicast VLAN Registration configuration
    network-clock       Show selector state.
    ntp                 Configure NTP
    perf-mon            Performance Monitor
    platform            Platform specific information
    port-security       Port security
    privilege           Display command privilege
    ptp                 Precision time Protocol (1588)
    pvlan               PVLAN configuration
    qos                 Quality of Service
    radius-server       RADIUS configuration
```

```
    rfc2544            RFC2544 perfomance tests
    rmon               RMON statistics
    running-config     Show running system information
    sflow              Statistics flow.
    snmp               Display SNMP configurations
    spanning-tree      STP Bridge
    switchport         Display switching mode characteristics
    tacacs-server      TACACS+ configuration
    terminal           Display terminal configuration parameters
    thermal-protect    Display thermal protection status.
    upnp               Display UPnP configurations
    users              Display information about terminal lines
    version            System hardware and software status
    vlan               VLAN status
    voice              Voice appliance attributes
    web                Web
```

Example 11. Using Context-sensitive Help for Discovery

The context-sensitive help feature for syntax display is useful as well while drilling down on the exact command to execute. In the following the user discovers the proper command 'show ip statistics system' through exploration:

```
my-device# show ip ?
    arp             Address Resolution Protocol
    dhcp            Dynamic Host Configuration Protocol
    http            Hypertext Transfer Protocol
    igmp            Internet Group Management Protocol
    interface       IP interface status and configuration
    name-server     Domain Name System
    route           Display the current ip routing table
    source          source command
    ssh             Secure Shell
    statistics      Traffic statistics
    verify          verify command

my-device# show ip statistics ?
    |             Output modifiers
    icmp          IPv4 ICMP traffic
    icmp-msg      IPv4 ICMP traffic for designated message type
    interface     Select an interface to configure
    system        IPv4 system traffic
    <cr>

! A repeated press of '?' displays the syntax:
my-device# show ip statistics ?
show ip statistics [ system ] [ interface vlan <v_vlan_list> ] [ icmp ]
[ icmp-msg <type> ]

my-device# show ip statistics system

IPv4 statistics:

  Rcvd:  2768 total in 181458 bytes
         1727 local destination, 0 forwarding
         0 header error, 0 address error, 0 unknown protocol
         0 no route, 0 truncated, 0 discarded
  Sent:  2553 total in 180047 bytes
         1512 generated, 0 forwarded
         0 no route, 0 discarded
  Frags: 0 reassemble (0 reassembled, 0 couldn't reassemble)
         0 fragment (0 fragmented, 0 couldn't fragment)
         0 fragment created
  Mcast: 0 received in 0 byte
```

```
         0 sent in 0 byte
  Bcast: 0 received, 0 sent
```

6.1 show running-config

'running-config' consists of a list of commands that, taken together, result in the currently running system configuration.

This list of commands is usually not 100% identical to the list of commands a user has input to configure the device. That is because 'running-config' is a *textual representation* of the system configuration which is stored in *binary form* in the RAM memory of the device.

Since the effective device configuration is huge, 'running-config' in the majority of cases only lists the delta between default settings and current settings. This significantly reduces the amount of output and greatly improves readability of the configuration, but it does require the reader to know what the default settings are.

It is possible, however, to also include values that are at default if the keyword 'all-defaults' is appended to the 'show running-config' command.

Example 12. Default vs. Non-default vs. All Defaults

In this example if the speed and duplex settings of an Ethernet interface are at default values (auto-negotiation) then nothing will be output. If the user then changes the speed to be fixed at 1Gbps then that value is now non-default and will be output. Duplex is also output, since it is forced to 'full' when the speed is fixed at 1Gbps.

```
! Display current configuration for an interface. All settings are at default:
my-device# show running-config interface GigabitEthernet 1/4
Building configuration...
interface GigabitEthernet 1/4
!
end

! Now set the speed to 1Gbps and display the configuration again:
my-device# configure terminal
my-device(config)# interface GigabitEthernet 1/4
my-device(config-if)# speed 1000
my-device(config-if)# end

my-device# show running-config interface GigabitEthernet 1/4
Building configuration...
interface GigabitEthernet 1/4
 speed 1000
 duplex full
!
end

! Include all default settings for that interface:
my-device# show running-config interface GigabitEthernet 1/4 all-defaults
Building configuration...
interface GigabitEthernet 1/4
 switchport voice vlan mode disable
 no switchport voice vlan security
 switchport voice vlan discovery-protocol oui
 loop-protect
 no loop-protect action
 loop-protect tx-mode
```

```
 switchport access vlan 1
 switchport trunk native vlan 1
 switchport hybrid native vlan 1
! ... much output omitted for brevity ...
```

Note how the output of 'show running-config' can be restricted to a specific interface. There are several other such filters, described below.

6.1.1 show running-config [all-defaults]

This displays the entire currently running system configuration.

6.1.2 show running-config feature *feature_name* [all-defaults]

Only output the commands relevant to a particular feature. The feature list depends on system configuration and software version. For example:

```
my-device# show running-config feature ?
    CWORD    Valid words are 'GVRP' 'access' 'access-list' 'aggregation'
             'arp-inspection' 'auth' 'clock' 'dhcp' 'dhcp-snooping'
             'dhcp_server' 'dns' 'dot1x' 'eps' 'erps' 'evc' 'green-ethernet'
             'http' 'icli' 'ip-igmp-snooping' 'ip-igmp-snooping-port'
             'ip-igmp-snooping-vlan' 'ipmc-profile' 'ipmc-profile-range' 'ipv4'
             'ipv6' 'ipv6-mld-snooping' 'ipv6-mld-snooping-port'
             'ipv6-mld-snooping-vlan' 'lacp' 'link-oam' 'lldp' 'logging'
             'loop-protect' 'mac' 'mep' 'monitor' 'mstp' 'mvr' 'mvr-port'
             'network-clock' 'ntp' 'perf-mon' 'phy' 'port' 'port-security'
             'ptp' 'pvlan' 'qos' 'rfc2544' 'rmon' 'snmp' 'source-guard' 'ssh'
             'thermal-protect' 'upnp' 'user' 'vlan' 'voice-vlan'
             'web-privilege-group-level'

my-device# show running-config feature dns
Building configuration...
!
vlan 1
!
!
!
ip dns proxy
!
interface GigabitEthernet 1/1
...
```

The structure of running-config is maintained in the output, i.e. sub-modes such as VLANs and Ethernet interfaces are listed but may be empty if the requested feature is irrelevant for the particular sub-mode.

6.1.3 show running-config interface *list* [all-defaults]

Show running-config for the specific list of Ethernet interfaces. This may contain wildcards, for example:

```
my-device# show running-config interface 2.5GigabitEthernet *
Building configuration...
interface 2.5GigabitEthernet 1/1
 speed 1000
 duplex full
!
```

```
interface 2.5GigabitEthernet 1/2
!
end
```

6.1.4 show running-config vlan *list* [all-defaults]

Show running-config for the specific list of VLANs, for example:

```
my-device# show running-config vlan 1-10
Building configuration...
vlan 1
 name default
!
end
```

In this example there is only one VLAN on the system.

6.1.5 show running-config interface vlan *list* [all-defaults]

Show running-config for the specific list of VLAN interfaces, for example:

```
my-device# show running-config interface vlan 1-10
Building configuration...
interface vlan 1
 ip address dhcp fallback 172.16.1.2 255.255.0.0
!
end
```

In this example there is only one VLAN interface on the system.

6.1.6 show running-config line { console | vty } *list* [all-defaults]

Show running-config for the console or list of virtual terminal devices (vty). On current designs there is a single console device, 0. Example:

```
my-device# show running-config line console 0
Building configuration...
line console 0
 exec-timeout 0 0
!
end
```

7 Working with Configuration Files

There are four kinds of configuration files:

- 'running-config', a virtual file containing the currently running system configuration
- 'startup-config', containing the boot-time configuration. When configuration is changed it must be copied to 'startup-config' in order to be applied at the next boot
- 'default-config', read-only and used when configuration is restored to defaults, i.e. also if 'startup-config' is missing. It contains product-specific customizations to the default settings of the device
- User-defined configuration files, of which there can exist up to two. These are typically used for backups or variants of 'startup-config'

All of these except 'running-config' are stored in the flash: file system. The available operations are:

copy *source destination*

The source and destination can be one of:

- running-config
- startup-config (or flash:startup-config)
- flash:*filename*
- tftp://*server*[:*port*]/*path-to-file*

dir

List the contents of the flash: file system

more flash: *filename*

Output the contents of the file to the terminal.

delete flash: *filename*

Delete the specific file.

7.1 Reverting to Default Configuration

It is possible to reset the total system configuration to defaults in two ways:

- Deleting 'startup-config' and rebooting
- Instructing the software to discard current configuration and reset to defaults without rebooting

Deleting 'startup-config' doesn't change 'running-config' until the system is rebooted, at which time defaults are loaded.

Conversely, discarding the current configuration does indeed affect 'running-config' but does not touch 'startup-config'. An explicit 'copy running-config startup-config' is necessary to make the change persistent.

Rebooting and resetting configuration to defaults is accomplished with the 'reload' command:

```
reload cold [ sid switch_id ]
reload defaults [ keep-ip ]
```

The first form reboots the system. If the system is stacking, a specific switch can be rebooted as well by supplying its switch ID.

The second form loads configuration defaults. If the 'keep-ip' keyword is given then the system attempts to keep the most relevant parts of the VLAN 1 IP setup in order to maintain management connectivity: The IP address setup and the active default route.

Note: There is no guarantee, however, that the above is sufficient. It depends on the actual network properties and the system's total IP configuration. In some cases it may be preferable to explicitly un-configure the system using 'no' commands, or prepare a suitable configuration and download it to the system's 'startup-config' and reboot.

Example 13. Working With Configuration Files

The following example assumes a file system which contains an additional file called 'backup', previously created with a 'copy' command.

```
! List files in flash:
my-device# dir
Directory of flash:
    r- 1970-01-01 00:00:00      648 default-config
    rw 1970-01-06 03:57:33     1313 startup-config
    rw 1970-01-01 19:54:01     1237 backup
3 files, 3198 bytes total.

! Display the contents of the file 'backup' (output is abbreviated):
my-device# more flash:backup
hostname my-device
...
end

! Use file 'backup' for the next boot by overwriting startup-config:
my-device# copy flash:backup startup-config
% Saving 1237 bytes to flash:startup-config

! Verify that the sizes are identical:
my-device# dir
Directory of flash:
    r- 1970-01-01 00:00:00      648 default-config
    rw 1970-01-06 05:30:41     1237 startup-config
    rw 1970-01-01 19:54:01     1237 backup
3 files, 3122 bytes total.

! Regret and delete startup-config. Note how 'flash:' is required:
```

```
my-device# delete flash:startup-config
my-device# dir
Directory of flash:
    r- 1970-01-01 00:00:00      648 default-config
    rw 1970-01-01 19:54:01     1237 backup
2 files, 1885 bytes total.

! Use the currently running config for next boot:
my-device# copy running-config startup-config
Building configuration...
% Saving 1271 bytes to flash:startup-config
```

Example 14. Using Reload Commands

```
! Reload defaults, but try to keep VLAN 1 configuration. First list current IP
! settings:
my-device# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.17/24       DHCP     UP

my-device# reload defaults keep-ip
% Reloading defaults, attempting to keep VLAN 1 IP address. Please stand by.
# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 172.16.1.17/24       DHCP     UP

! Contents of flash: are unchanged:
my-device# dir
Directory of flash:
    r- 1970-01-01 00:00:00      648 default-config
    rw 1970-01-06 05:33:18     1237 startup-config
    rw 1970-01-01 19:54:01     1237 backup
3 files, 3122 bytes total.

! Reload again, but don't try to keep VLAN 1 settings:
# reload defaults
% Reloading defaults. Please stand by.

! Verify that the default IP settings have been restored:
# show ip interface brief
Vlan Address              Method   Status
---- -------------------- -------- ------
   1 192.0.2.1/24         Manual   UP

! Reboot the system
# reload cold
% Cold reload in progress, please stand by.
! ... bootup output omitted ...
```

8 Working with Software Images

The system can store up to two software images that are stored in FLASH. The image selected for bootup is termed the Active image, while the other is termed the Alternate image.

It is possible to *swap* the Active and Alternative image, and it is possible to *upgrade* to a new Active image.

A swap simply switches the Active/Alternate designation on each image and reboots the system.

A firmware upgrade performs these steps:

- Download new firmware using TFTP and verify suitability for the system
- Overwrite the current Alternate image with the newly downloaded image
- Swap Active/Alternate and reboot

The result is that the old Active build becomes Alternate, and the newly downloaded image Active.

The relevant commands are:

show version
firmware swap
firmware upgrade tftp://*server*[:*port*]/*path_to_file*

'show version' lists various details about the system, including the images in FLASH.