ADTRAN Switch Engine (ASE)

CLI Configuration

Configuration Guide

6AMCCG0001-29C

June 2020

To the Holder of this Document

This document is intended for the use of ADTRAN customers only for the purposes of the agreement under which the document is submitted, and no part of it may be used, reproduced, modified or transmitted in any form or means without the prior written permission of ADTRAN.

The contents of this document are current as of the date of publication and are subject to change without notice.

Trademark Information

“ADTRAN” and the ADTRAN logo are registered trademarks of ADTRAN, Inc. Brand names and product names included in this document are trademarks, registered trademarks, or trade names of their respective holders.

Disclaimer of Liability

The information or statements given in this document concerning the suitability, capacity, or performance of the mentioned hardware or software products are given “as is”, and any liability arising in connection with such hardware or software products shall be governed by ADTRAN’s standard terms and conditions of sale unless otherwise set forth in a separately negotiated written agreement with ADTRAN that specifically applies to such hardware or software products.

To the fullest extent allowed by applicable law, in no event shall ADTRAN be liable for errors in this document for any damages, including but not limited to special, indirect, incidental or consequential, or any losses, such as but not limited to loss of profit, revenue, business interruption, business opportunity or data, that may arise from the use of this document or the information in it.

Revision History

Rev A	March 2019	Initial Release
Rev B	July 2019	Removed OSPF from the context-sensitive help command output example.
Rev C	May 2020	Updated format and supported hardware.

Table of Contents

1. Overview

This configuration guide describes basic usage and configuration of the Command Line Interface (CLI), a comprehensive management interface for use with ADTRAN Switch Engine (ASE) products. While CLI describes the method used to communicate, such as by console or Telnet, it also refers to the way information is passed to the unit. As a text-based user interface, the CLI prompts you to input commands line by line when you interface with the ASE unit (hence the name command line interface).

CLI is the only management interface accessible on the serial console. Even without network connectivity, ASE devices can be managed using a serial connection.

2. Hardware and Software Requirements and Limitations

ASE CLI Configuration is supported on the ASE products outlined in *Table 1* below:

Table 1. Supported Products

Product	P/N
NetVanta 1560-08-150W Switch	17108108PF2
NetVanta 1560-24-740W Switch	17108124PF2
NetVanta 1560-48-740W Switch	17108148PF2
NetVanta 1560-08-65W Switch	17101561PF2
NetVanta 1560-24-370W Switch	17101564PF2
NetVanta 1560-48-370W Switch	17101568PF2

3. CLI Overview

Introduction to Commands

The most important part is understanding that commands make the ASE unit function. The right commands lead to a fully functioning unit, whereas improperly entered or forgotten commands prevent the unit from functioning. To properly use commands, you must understand what function you want the ASE unit to complete and what syntax the unit understands as instructions. Each command has its own role within the operating system, and it is the responsibility of the operator to become familiar with specific commands and command sets.

The following are key characteristics of the CLI:

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

The CLI can be accessed directly using the serial console, or over the network through telnet or ssh. In each case, the user has to log in before CLI commands can be executed. This begins a session that lasts until logout.

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, such as 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.

How Commands Function

A command is a single line of syntax composed of two main parts. The most important part is the command itself, or the command word. Most command words are short and straightforward (for example, do, exit, or configure). Command words are entered immediately after the command prompt in the CLI.

The second part of a command is its argument. An argument is a specification that modifies the command. In the command **show version**, **show** is the command word and **version** is the argument because it modifies the command **show**. Commands can have any number of arguments, depending upon the action required of the unit, and in some instances you have a choice of arguments to use.

Optionally, some commands use variables to specify information relevant only to your ASE unit. These variables are identified with angled brackets (< >). The description of the information required is contained within the symbols and displayed in italics. For example, the following command provides the command word **hostname** and includes the variable <*host_name*>:

hostname *<host_name>*

> *Command words are not case sensitive, thus show, SHOW, and Show are identical. Conversely, parameters may either be case-sensitive or not, depending on the command and parameter in question.*

ASE Command System

ADTRAN products, training tools, and manuals follow a specific system for entering and referencing commands. Items that are typed in **bold** are the required commands and arguments for a certain action. In the following documentation, you will see commands in bold after an example prompt. They look similar to this:

```
>enable
# configure terminal
(config)# line vty 1
(config-line)#
```

In the example above, the characters >, #, (config)#, and (config-line)# are the prompts after which commands are entered. In this example, the words in bold (**enable, configure terminal,** and **line vty 1**) are the entire commands and constitute what should be typed after the prompt. It is important to pay attention to the prompt you are given when communicating with your unit, because certain commands only work in certain modes, which are signified by the prompt.

In certain commands, you are given a choice of arguments. If this is the case, the manual or guide will place the argument in brackets separated by a vertical bar (|) between your choices as seen in this example:

```
(config)# clock [summer-time | timezone]
```

Again, remember the # is your prompt, the command word is clock, and your choices of arguments are **summer-time**, and **timezone**.

Certain commands require you to enter your own information which are called variables. Information within a command line that pertains to your personal unit is set off with angled brackets (< >). The description of the information required is contained within the angled brackets and is displayed in *italics*. For example:

```
# hostname <host_name>
```

In this case, # is your prompt, the command word is **hostname**, and the information needed from you is the name of the host (unit) *(<host_name>)*.

Understanding Command Modes

As you begin communication, you should understand the command modes. Just as there are different levels of commands in the CLI, there are different modes for commands within ASE itself. Each command mode enables the user to access more commands, and make more changes in the unit's configuration.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 login. See *Understanding Privilege Levels on page 11* for more information on privilege levels.

The ASE CLI has three main command modes: Basic, Enable, and Global Configuration. These command modes are organized in a three-tiered hierarchy with Basic at the bottom, then Enable, and Global Configuration at the top. In addition, configuration command sets are available from Global Configuration mode. These command sets are broken down into categories of similar functions. For example, all commands pertaining to configuring the interfaces are grouped together.

Basic Mode

Interaction with your unit begins at the Basic mode. The commands supported at this command tier are limited, as is interaction with the unit itself. The Basic mode prevents users without access to the higher tiered commands from changing the preferred configurations of the unit. *Table 2* describes the Basic mode.

Table 2. Basic Mode

Mode	Access By...	Mode Prompt	Accessible Commands
Basic	Beginning an ASE session	>	■ Display system information ■ Perform traceroute and ping functions

Enable Mode

Enable mode is one step up from the Basic mode. ADTRAN suggests that a password be required to access the Enable mode. Refer to the quick start guides shipped with your unit and located online at https://support-forums.adtran.com for more information on configuring a password.

From the Enable mode, you can access the configurations of your product, as well as handle how your unit boots and runs, among other things. *Table 3* describes the Enable mode.

Table 3. Enable Mode

Mode	Access By...	Mode Prompt	Accessible Commands
Enable	Entering **enable** while in the Basic mode as follows: **>enable**	#	■ Manage the startup and running configurations ■ Enable and disable **debug** commands ■ View **show** command output ■ Enter any of the configuration modes

Global Configuration Mode

The Global Configuration mode is the highest level tier within ASE. The Global Configuration mode allows the user to make changes regarding the entire product system. All of your system's configurations are accessed through the Global Configuration mode. From this level, you can access not only line configurations, router configurations, and interface configurations, but also any other configurations or parameters on your system. *Table 4* describes the Global Configuration mode.

Table 4. Global Mode

Mode	Access By...	Mode Prompt	Accessible Commands

Global Config	Entering **config** while at the Enable mode as follows:
	`>enable`
	#
	`#config t`
		`(config)#`	■ Set the system's Enable-level password(s)
			■ Configure the system global IP parameters
			■ Configure the SNMP parameters
			■ Enter any of the configuration modes

Configuration Command Sets

Configuration Command Sets contain a group of commands that are specific to a particular interface, feature, protocol, etc. For example, there are specific command sets for VLANs, Ethernet interfaces, and IP. *Table 5* summarizes some of the command sets available in the ASE CLI. These command sets are available from Global Configuration Mode.

Table 5. Configuration Command Sets Summary

Command Set	Acceessed By...	Description
DHCP Pool Config	Specifying a Dynamic Host Configuration Protocol (DHCP) client pool name as follows: `(config)#ip dhcp pool` *<name>*	Configure DHCP client pools Prompt: hostname(config-dhcp-pool)#
Gigabit Ethernet Interface	Specifying a Gigabit Ethernet interface number at the Global Configuration mode prompt as follows: `(config)#interface gigabitethernet` *<port_type_list>*	Configure Gigabit Ethernet interfaces Prompt: hostname(config-if)#
IPMC Profile	Specifying an IP Multicast profile name as follows: `(config)#ipmc profile` *<name>*	Configure IP Multicast profiles Prompt: hostname(config-ipmc-profile)#
JSON Notification Host Config	Specifying a JavaScript Object Notation (JSON) RPC name as follows: `(config)#json notification host` *<name>*	Configure JSON notification hosts Prompt: hostname(config-json-notif-host)#
Line	Specifying a terminal line number, console terminal line number, or virtual terminal line number as follows: `(config)#line [`*<number>* `| console 0 | vty` *<number>*`]`	Configure terminal lines Prompt: hostname(config-line)#

Table 5. Configuration Command Sets Summary *(Continued)*

Command Set	Accessed By...	Description
SNMP Server Host	Specifying a SNMP server host name as follows: `(config)#snmp-server host <name>`	Configure SNMP server host entries Prompt: hostname(config-snmps-host)#
STP Aggregation Config	Specifying Spanning Tree Protocol Aggregation mode as follows: `(config)#spanning-tree aggregation`	Configure Spanning Tree Protocol aggregation Prompt: hostname(config-stp-aggr)#
VLAN	Specifying a VLAN ID at the Global Configuration mode prompt as follows: `(config)#vlan <vlan_list>`	Configure active VLANs Prompt: hostname(config-vlan)#
VLAN Interface	Specifying a VLAN interface number at the Global Configuration mode prompt as follows: `(config)#interface vlan <vlan_list>`	Configure VLAN interfaces Prompt: hostname(config-if-vlan)#

Command Mode Transitions

A user can transition between command modes and configuration command sets, subject to the user's privilege level and the current session privilege level (see *Understanding Privilege Levels on page 11*).

The Basic Mode privilege level can be raised 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 Enable mode, it is possible to enter into Global Configuration mode by entering the command **configure terminal**. Exit from Global Configuration mode is achieved by entering the command **end** or **exit** or pressing Ctrl-Z.

Access to a configuration command set (for example, Ethernet interfaces) goes through Global Configuration or another command set. Thus, it is possible to change directly from the VLAN command set to the Gigabit Ethernet interface command set.

Each mode and command set implements a scope for commands. Inside each mode, a particular subset of commands is available. To access other commands, one must generally change modes/command sets. This change is necessary because there are commands with identical prefixes in different modes. For example, there are commands that begin with 'ip' in Enable, Global Configuration, and VLAN Interface Configuration modes.

There is one exception to this: Enable mode commands (whether privileged or unprivileged) are accessible from within Global Configuration or one of the command sets by using the **do** command. See *Using the do Command to Execute Commands Across Modes on page 11*.

Figure 1 shows the possible transitions between three major command modes and command sets, and some of the relevant commands.

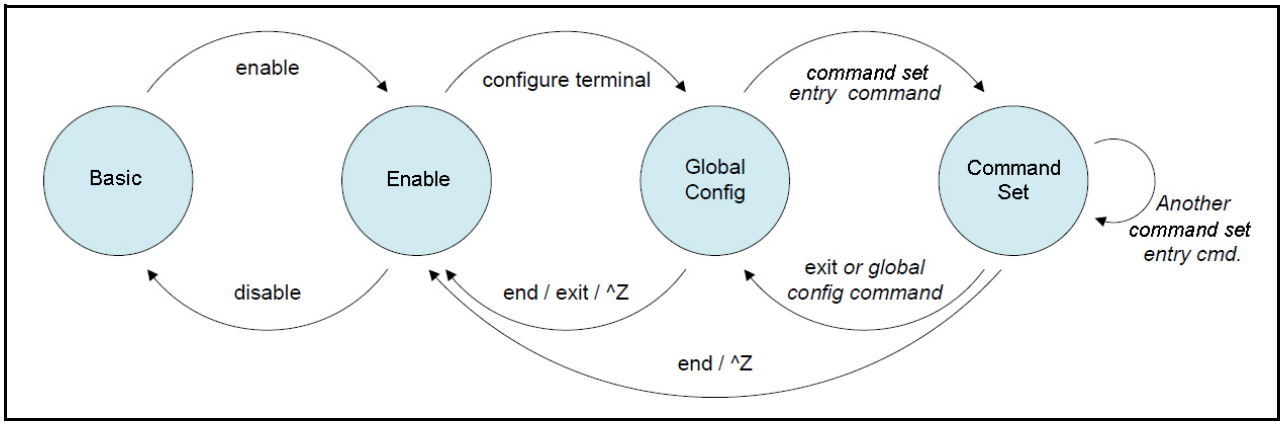

Figure 1. Command Mode Transitions

Changing Between Command Modes

! Initial mode for this example is Basic. Raise level
! (and change mode):

```
my-device> enable
Password: ***
my-device#
```

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

! Enter Global Configuration mode:

```
my-device# configure terminal
```

! Now create VLAN 100 and give it a name. This enters the VLAN command set, 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 Enable

```
my-device(config)# exit
```

! And use 'disable' to go back to Basic:

```
my-device# disable
my-device>
```

Using the do Command to Execute Commands Across Modes

The **do** command provides a way to execute commands in other command sets without having to exit the current command set. In the following example, the user wants to change the IP address on the VLAN 1 interface and uses **do** to verify the current address while in the VLAN command set:

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

```
my-device(config-if-vlan)# end
! When in Enable, no 'do' prefix is needed:
my-device# show ip interface brief
Interface             Address              Method    Status
---------------       --------------------------    ---------    ------
VLAN 1                172.16.1.15/24       DHCP      UP
```

Other Special Keys

Table 6 highlights one additional key that is defined as a convenience. It allows the immediate return from any command set to Enable mode.

Table 6. Special Keys

Key	Operation
CTRL-Z	Return directly to Enable mode

Understanding Privilege Levels

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

Each user on the device has a default privilege level that is copied to the session's privilege level at login. 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 reverts 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 inputs a new password, as the operator will usually not know the encrypted form of the password.

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

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 Enable 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)#
```

4. Quickstart

This section describes how to perform the following:

- Connecting to the CLI
- Logging in and resetting configuration to factory defaults
- Setting device hostname and admin user password
- Setting VLAN 1 IP address
- Verifying connectivity using 'ping'
- Displaying the current configuration and save it to flash storage

Connecting to the CLI

The device should be powered on and have a functional connection to a computer using the serial console port on the device. The computer must be running a VT100 terminal emulation program such as TeraTerm or PuTTY on Windows, or Minicom on Linux.

Once you have connected to the unit, adjust the program settings as follows:

- 115200 baud
- No parity
- 8 data bits
- 1 stop bit

- No flow control

If you are using a VT100 terminal emulation program, name your new connection and set up the new connection. Verify COM 1 is the type of connection you are using. Once you have entered the program settings and applied them, you should be presented with a terminal window with which to interface with your unit.

Logging In and Resetting Configuration to Factory Default

Press ENTER one or more times until the Username: prompt appears. Type **admin** and press ENTER. At the **Password:** prompt type **password** (this is the default password) and press ENTER. This completes the login sequence and displays the prompt, '#'.

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

> *The prompt will not display the password as you type it.*

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. Enter the command **reload defaults**. When the prompt returns, the system has reverted to factory defaults as follows.

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

Setting Device Hostname and Admin User Password

The CLI has several different modes. The current mode is called enable 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 items. Such operations are performed while in the Global Configuration mode.

To set the device hostname, first change to Global Configuration mode by entering the command **configure terminal,** then type **hostname** <*name*>, where <*name*> is a suitable name for the device. Exit the configuration menu. The sequence should appear as shown here (**my-device** is the hostname used in this example).

```
# 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. A password should be set for the admin user.

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

The user, admin, now has the password *verysecret*. Other users can be added in similar fashion.

Setting 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) or static IP address allocation.

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 properly 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 (DNS) server addresses.

The configuration proceeds in the same manner as setting the hostname: Enter configuration mode, input and execute configuration commands, leave configuration mode. The following commands instruct the device to use DHCP to obtain an IP address, or, if DHCP fails, to use a static fallback address. Inclusion of a fallback IP is optional and may be omitted.

```
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. As seen below, the DHCP negotiation succeeded and the device obtained an address:

```
Interface             Address              Method    Status
---------------       --------------------------    ---------    ------
VLAN 1                172.16.1.15/24       DHCP      UP

my-device#
```

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

If DHCP negotiation failed, then the fallback IP of 172.16.1.2/255.255.0.0 would be assigned.

Now, the most basic system configuration is complete. Management connectivity can be verified by issuing a **ping** command to a known working internal IP address:

```
my-device# ping ip 172.16.1.1
PING 172.16.1.1 (172.16.1.1): 56 data bytes.
64 bytes from 172.16.1.1: seq=0 ttl=116 time=25.837 ms
64 bytes from 172.16.1.1: seq=1 ttl=116 time=22.113 ms
64 bytes from 172.16.1.1: seq=2 ttl=116 time=21.702 ms
64 bytes from 172.16.1.1: seq=3 ttl=116 time=26.034 ms
64 bytes from 172.16.1.1: seq=4 ttl=116 time=24.285 ms

--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 21.702/23.994/26.034 ms
```

If the ping is successful, network logins can now be performed through telnet or ssh to the address on VLAN interface 1, 172.16.1.15 (or 172.16.1.2).

Displaying and Saving Configuration to Flash

The current configuration of the ASE device can be displayed in the form of a virtual file containing the full set of commands necessary to create an identical configuration. A few exceptions exist because certain items, such as private SSH keys, are not displayed, This file is called running-config and is volatile by nature; it does not survive across reboots. Therefore, it is necessary to save the file to flash storage under the name *startup-config*, as a file in flash storage is read and executed upon every boot and is responsible for restoring the running configuration of the system to the state of the unit when the last save took place.

Displaying the Running Configuration

The command **show running-config** will display the configuration settings as seen in the following sample output.

> *Some details were edited for brevity. The set of interfaces displayed in the output is dependent on hardware capabilities.*

```
my-device# show running-config
Building configuration...
hostname my-device
username admin privilege 15 password encrypted
3ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e0341724015282297
95a5c9529dfbc04c86e01
!
vlan 1,42
!
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
!
line vty 0
!
! [...]
!
end
my-device#
```

Lines that begin with '**!**' are comments. The file begins with the **hostname** command and the password for the admin user, followed by VLANs 1 and 42 and other items, such as Spanning Tree Protocol (STP). A list of all port interfaces on the device, ordered by type, and port number comes next.

Since the port interfaces are set to default, there is no additional data to list. As a general rule, only non-default configuration is displayed, otherwise the output would be large and readability would suffer. There are a few exceptions that will be discussed later.

Following the physical interfaces is VLAN interface 1. When the IP address is missing, it will not list it in this section. Finally, the line section is shown. It specifies characteristics for the serial console (line console 0) or network CLI management connections (line vty x).

The configuration as displayed above is also what is saved to *startup-config*.

Saving the Running Configuration

The command **copy running-config startup-config** will save the running configuration to flash storage as seen in the following sample output:

```
my-device# copy running-config startup-config
Are you sure you want to save the configuration (yes/no)? yes
Building configuration...
% Saving 2796 bytes to flash:startup-config
my-device# dir
Directory of flash:
     r- 2018-08-02 07:11:49      650 default-config
     rw 1970-01-01 00:02:19      3478 startup-config
2 files, 4128 bytes total.

Flash size: 8687616 bytes (8.3 MiB)
Flash free: 8654848 bytes (8.3 MiB)
my-device# more flash:startup-config
hostname my-device
username admin privilege 15 password encrypted
3ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e0341724015282297
95a5c9529dfbc04c86e01
!
vlan 1,42
[...]
```

The **dir** command lists the files in the flash file system while **more** outputs the contents of the designated file.

The skills exercised in this section form the basis for all day-to-day work with the CLI on the device: logging in, displaying information with the **show** command, working with configuration files (**show running-config, copy, dir, more**), working with the actual configuration (**configure terminal, exit**), and sub-modes (**interface** ...).

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

5. CLI Basics

This section highlights some of the common shortcuts and abbreviations available within the CLI. Also included are basic tasks with example configurations.

Abbreviating Keywords and Parameters

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:

- There are many keywords that begin with *s* but only one that begins with *sh*
- There are several commands that begin with *show i* but only one that begins with *show in*
- The **show interface** command takes a port type as parameter. Since the ASE unit only has GigabitEthernet port types, *g* is a unique abbreviation for GigabitEthernet
- 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
- 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 CLI (see *Context-Sensitive Help on page 19*).

Using the Keyboard

The CLI 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

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. *Table 7* shows the available editing functions and keys.

Table 7. Basic Line Editing Key

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

Table 7. Basic Line Editing Key *(Continued)*

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

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 8* lists the keys and their operation when viewing command history.

Table 8. Command History

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 between 0 and 32, where 0 disables the history altogether.

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

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.

Context-Sensitive Help

The CLI implements several hundred commands ranging from the very simple to the very complex. Therefore, it is 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 9* lists the keys and their operation for viewing context-sensitive help.

Table 9. Context-Sensitive Help

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 if it is unambiguous

The context-sensitive help only displays commands that are accessible at the current session privilege level (see *Understanding Privilege Levels on page 11*).

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
   alarm           alarm
   auto-link       Display Auto-link Status
```

! The same, but without descriptions:

```
my-device# show a<TAB>
aaa              access      access-list     aggregation       alarm     auto-link
```

! 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 ]
```

Long Lines and Pagination

A session has a configuration that indicates the width of the terminal in characters and the length 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, see *Understanding Terminal Parameters on page 22*.

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 the 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
```

Table 10 outlines the keys that control pagination.

Table 10. Pagination Keys

Key	Operation
ENTER	Display next line of output
SPACE	Display next page of output
G	Display remainder of output without more pagination

Table 10. Pagination Keys *(Continued)*

Key	Operation
Q/CTRL-C	Display remainder of output
Any other key	Display next page of output. Certain terminal keys (arrows, Home, End, etc.) may appear as multiple characters to the CLI, 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
```

The same is true for setting the terminal width in characters.

Filtering Output

The output from commands can be filtered in most cases. 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.
```

Understanding Terminal Parameters

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

Table 11 lists available settings and modes where each can be configured.

Table 11. Setting and Modes

Setting	Modes	Description
editing	Enable, Line	Enable/disable command line scrolling
exec-banner	Line	Enable/disable display of the Exec banner (configured with 'banner exec ...')
exec-timeout	Enable, Line	Inactivity timer; automatically log out after a period of inactivity. A value of zero disables automatic logout
history	Enable, Line	Length of command history buffer
length	Enable, Line	Terminal length in lines, used for pagination. Zero disables pagination
location	Line	A line of text that describes the terminal location (such as "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	Enable, 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 vty is an abbreviation for Virtual TTY and X is a value between 0 and 15.

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.

Changing Terminal Parameters

This example shows how to change some 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
Are you sure you want to save the configuration (yes/no)? yes
```

```
Building configuration...
% Saving 1287 bytes to flash:startup-config
my-device#
```

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 these banners are configured in a similar manner, using the **banner** command:

banner [motd] *<delimeter> <message> <delimeter>*

banner exec *<delimeter> <message> <delimeter>*

banner login *<delimeter> <message> <delimeter>*

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. The delimiters are not included in the actual text.

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.
% Entering multi-line text input mode. Type in text and exit the mode using the
delimiting character '*'. All input after that character will be silently ignored.
The effective buffer size, i.e. excluding the delimiting characters but including
any newline characters (e.g. from multi-line input), cannot be longer than 255.
my-device(multiline-input)# It spans multiple lines.
my-device(multiline-input)# And one more. But now it ends.*


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


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
Are you sure you want to save the configuration (yes/no)? yes
Building configuration...
% Saving 1461 bytes to flash:startup-config
my-device#
```

Ethernet Interface Naming

An Ethernet interface, or port, is identified by two pieces of information:

- The type (GigabitEthernet)
- The ID of the switch it belongs to (this value is always 1)
- The port number within the type and switch (**GigabitEthernet 1/1**)

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

It is possible to *wildcard* the port to mean “all ports”. A wildcard is written with an asterisk instead of type or port:

- * means all ports of all types on the switch
- *type* * means all ports of the specified type on the switch

6. Configuring the System

Changes to system configuration can only be made from the Global Configuration mode and the command sets, except when working with configuration files or reloading defaults. This is done in Enable mode. The following steps outline the sequence.

1. Raise privilege level to 15.
2. Enter Global Configuration mode.
3. Input appropriate configuration commands. Optionally, enter command sets and input appropriate commands there.
4. Exit Global Configuration mode.
5. Verify configuration.
6. Save configuration to flash.

Configuration Example

1. Raise privilege level:

```
>enable
Password: ***
```

2. Enter Global Configuration mode:

```
#configure terminal
```

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
```

4. Leave Global Configuration mode and go back to Enable:

```
my-device(config)#end
```

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
3ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e0341724015282297
95a5c9529dfbc04c86e01
!
vlan 1
  name default
!
interface GigabitEthernet 1/1
!
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
Interface             Address             Method    Status
---------------       --------------------------    ----------    ------
VLAN 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
```

6. Save configuration to flash:

```
my-device#copy running-config startup-config
Are you sure you want to save the configuration (yes/no)? yes
Building configuration...
% Saving 1272 bytes to flash:startup-config
```

Resetting or Removing Configuration with *no*

It is possible to remove specific configuration items or reset them to their default values. In general, almost every 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.

Using *no* Forms

The following list shows the tasks accomplished:

- Configure the VLAN 1 interface IP address to use DHCP
- Configure the DNS name server to be taken from DHCP
- Inspect the configuration
- Remove the DNS name server
- Remove 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 no 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
```

```
Interface      Address                 Method    Status
--------------     -------------------------     ---------    ------
VLAN 1         172.16.1.15/24          DHCP      UP

my-device# show ip name-server
Configured DNS server 0 is set by DHCPv4:
172.16.1.1 is used for DNS lookup.
Configured DNS server 1 is set by NONE:
No address is used for DNS lookup.
Configured DNS server 2 is set by NONE:
No address is used for DNS lookup.
Configured DNS server 3 is set by NONE:
No address is used for DNS lookup.
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
Configured DNS server 0 is set by NONE:
No address is used for DNS lookup.
Configured DNS server 1 is set by NONE:
No address is used for DNS lookup.
Configured DNS server 2 is set by NONE:
No address is used for DNS lookup.
Configured DNS server 3 is set by NONE:
No address is used for DNS lookup.
```

> *The syntax of the configuration commands and their 'no' forms are different; the 'no' forms usually do not take as many parameters*

This is usually convenient but may give surprising results in certain cases. For example, an OAM MEP instance can configure Continuity Check using the command **mep** <*number*> **cc** <*priority number*> and reset it with **no mep** <*number*> **cc,** where the same MEP instance number is used for <*number*>**.** However, because MEPs are removed using the command **no mep** <*number*>, it is possible to unintentionally remove an existing MEP by entering **no mep** <*number*> **ccc** – the extra 'c' means that the last word isn't recognized as 'cc', leading to a match of the MEP removal command instead of the desired reset-CC command.

7. Managing 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>
```

The command **password none** is used when no password is desired. The security implications of using this should be considered carefully. Likewise, **no username** deletes the given user account.

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
3ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e0341724015282297
95a5c9529dfbc04c86e01

! 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 asecret
my-device(config)# username monitor privilege 1 password unencrypted newsecret

! 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
3ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e0341724015282297
95a5c9529dfbc04c86e01
username operator privilege 10 password encrypted
015b1985b585ab353c37a4441e034172401528229795a5c9529dfbc04c86e012138abcd88dda222affe
a861485e60e6407c5a328
username monitor privilege 1 password encrypted
7cc9861485e60e6407c56a16a7cc9861485e60e6407c5a328441e034172401528229795a5c95229795a
5c9529dfbc04c86e01abc

! 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
ad61dc090116a16a7cc9861485e60e6407c5a328015b1985b585ab353c37a4441e03417240152822979
5a5c9529dfbc04c86e01
username monitor privilege 1 password encrypted
7cc9861485e60e6407c56a16a7cc9861485e60e6407c5a328441e034172401528229795a5c95229795a
5c9529dfbc04c86e01abc
```

8. Using Show Commands

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

> *The exact set of available commands, parameters, and output format depends on the system configuration and software version.*

The **show** commands exist only in Basic and Enable 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.

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:

```
my-device>enable
Password:***
my-device#show ?
aaa                         Authentication, Authorization, and Accounting methods
access                      Access management
access-list                 Access list
aggregation                 Aggregation port configuration
alarm                       alarm
auto-link                   Display Auto-link Status
board-data                  Model Name
clock                       Configure time-of-day clock
ddmi                        DDMI configuration
dot1x                       IEEE Standard for port-based Network Access Control
eps                         Ethernet Protection Switching
erps                        Ethernet Ring Protection Switching
green-ethernet              Green Ethernet (Power reduction)
history                     Display the session command history
interface                   Interface
ip                          Interface Internet Protocol configuration commands
ipmc                        IPv4/IPv6 multicast configuration
ipv6                        IPv6 configuration commands
lacp                        LACP configuration/status
licenses                    Display license information.
line                        TTY line information
link-oam                    Link OAM configuration
lldp                        Link Layer Discover Protocol
logging                     System logging message
loop-protect                Loop protection configuration
mac                         Mac Address Table information
mep                         Maintenance Entity Point
monitor                     Monitoring different system events
mrp                         MRP status
mvr                         Multicast VLAN Registration configuration
ntp                         Configure NTP
platform                    Platform configuration
poe                         Power Over Ethernet
port-security               Show Port security overview status
```

```
privilege                   Display command privilege
process                     process
ptp                         Precision time Protocol (1588)
pvlan                       PVLAN configuration
qos                         Quality of Service
radius-server               RADIUS configuration
rmon                        RMON statistics
running-config              Show running system information
sflow                       Statistics flow.
snmp                        Set SNMP server's configurations
spanning-tree               STP Bridge
svl                         Shared VLAN Learning configuration
switchport                  Display switching mode characteristics
system                      system
tacacs-server               TACACS+ configuration
terminal                    Display terminal configuration parameters
thermal-protect             Display thermal protection status.
udld                        Unidirectional Link Detection (UDLD) configura-
                            tions,statistics and status
upnp                        Display UPnP configuration
user-privilege              Users privilege configuration
users                       Display information about terminal lines
version                     System hardware and software status
vlan                        VLAN status
voice                       Voice appliance attributes
web                         Web
```

Using Context-Sensitive Help for Discovery

The context-sensitive help feature for syntax display is also useful for determining the exact command to execute. In the following example, the user discovers the proper command **show ip statistics** system through exploration:

```
my-device# show ip ?
   acd                    Address Conflict Detection
   arp                    Address Resolution Protocol
   dhcp                   Dynamic Host Configuration Protocol
   domain                 Default domain name
   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
system                   IPv4 system traffic
<cr>
```

```
! A repeated press of '?' displays the syntax:
my-device# show ip statistics ?
show ip statistics [ system ]
my-device# show ip statistics system

IPv4 system statistics:
Rx Packets:                  1378216        Tx Packets:            1279763
Rx Octets:                 136914382        Tx Octets:           129975556
Rx Unicast:                  1280936        Tx Unicast:            1279763
Rx Multicast:                      0        Tx Multicast:                0
Rx Broadcast:                  97280        Tx Broadcast:                0
Rx Discards:                       0        Tx Discards:                 0
Rx ReasmOKs:                       0        Tx FragOKs:                  0
Rx ReasmReqds:                     0        Tx FragCreates:              0
Rx ReasmFails:                     0        Tx FragFails:                0
Rx Delivers:                 1378076
Rx HdrErrors:                      0
Rx AddrErrors:                    60
Rx UnknProtos:                     0
Rx Truncated:                      0
```

Show running-config

The virtual file *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 that is stored in binary form in the RAM memory of the device.

Because 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.

With **show running-config all-defaults**, it is possible to include values that are at default.

Default vs. Non-default vs. All Defaults

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

```
! 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
...
```

The output of **show running-config** can be restricted to a specific interface. There are several such filters, described in the next sections.

Show running-config [all-defaults]

This displays the entire currently-running system configuration.

Show running-config interface list [all-defaults]

By using this filter, the user can review a specific list of Ethernet interfaces. This may contain wildcards, for example:

```
my-device# show running-config interface GigabitEthernet *
Building configuration...
interface GigabitEthernet 1/1
   speed 1000
   duplex full
!
interface GigabitEthernet 1/2
```

```
!
interface GigabitEthernet 1/3
!
interface GigabitEthernet 1/4
!
interface GigabitEthernet 1/5
!
interface GigabitEthernet 1/6
!
interface GigabitEthernet 1/7
!
interface GigabitEthernet 1/8
!
interface GigabitEthernet 1/9
!
interface GigabitEthernet 1/10
!
end
```

Show running-config interface vlan list [all-defaults]

It is also possible to filter the list of VLAN interface, 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.

Show running-config line [console | vty] list [all-defaults]

This command can be used for the console or list of virtual terminal devices (vty). On current designs, there is a single console device, 0. For example:

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

9. Working with Configuration Files

There are four kinds of configuration files:

- running-config – a virtual file containing the currently running system configuration.
- startup-config – contains 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 – a read-only file used when configuration is restored to defaults. This file is also used if *startup-config* is missing. It contains product-specific customizations to the default settings of the device.

- User-defined – configuration files created by the user (up to 31). 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>*

where <*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

Outputs the contents of the file to the terminal.

delete flash:filename

Erases the specific file.

Reverting to Default Configuration

It is possible to reset the system to a default configuration in two ways:

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

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

Conversely, discarding the current configuration affects *running-config* but not *startup-config*. An explicit **copy running-config startup-config** is necessary to make the change persistent.

Rebooting and resetting the default configuration is accomplished with the **reload** command:

reload cold

reload defaults [keep-ip]

The reload cold version reboots the system.

The second method 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).

There is no guarantee, however, that the above is sufficient for reverting to default configuration: 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.

Booting from a Backup File

The following example assumes a file system that 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
Are you sure you want to save the configuration (yes/no) ? yes
% Saving 1791 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
```

! 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
Are you sure you want to save the configuration (yes/no) ? yes
% Saving 1271 bytes to flash:startup-config
```

Using Reload Commands

! Reload defaults, but try to keep VLAN 1 configuration. First list current IP
! settings:

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

```
my-device# reload defaults keep-ip
% Reloading defaults, attempting to keep VLAN 1 IP address. Please stand by.
# show ip interface brief
Interface      Address               Method     Status
---------------      --------------------------     ----------      ------
VLAN 1         172.16.1.15/24        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
Interface       Address                 Method      Status
--------------      -------------------------     ---------     ------
VLAN 1          172.16.1.15/24          DHCP        UP
```

! Reboot the system

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

10. Working with Software Images

The system can store up to two software images 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 the Alternative image, and it is possible to upgrade to a new Active image. A swap simply switches the Active and Alternate designation on each image and reboots the system.

A firmware upgrade performs these steps:

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

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

The relevant commands are:

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

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

11. Warranty and Contact Information

Warranty

Warranty information can be found online by visiting www.adtran.com/warranty.

Contact Information

To contact ADTRAN, choose one of the following methods:

Department	Contact Information	
Customer Care	From within the U.S.: From outside the U.S.:	(888) 4ADTRAN ((888)-423-8726)+ +1 (256) 963-8716
Technical Support	Support Community Product Support:	www.supportforums.adtran.com www.adtran.com/support
Training	Email: ADTRAN University:	training@adtran.com www.adtran.com/training
Sales	For pricing and availability:	1 (800) 827-0807