

Chapter 2

Using the Command Line Interface (CLI)

| | |
|---|------|
| Introduction | 2-2 |
| Parts of a Command | 2-2 |
| Entering a Command | 2-3 |
| Shortcuts | 2-3 |
| Command Sets | 2-5 |
| What is Valid Syntax? | 2-6 |
| What Commands does a Feature Support? | 2-8 |
| Editing Commands | 2-9 |
| Recalling Commands | 2-9 |
| Command Reference | 2-11 |
| add alias | 2-11 |
| delete alias | 2-13 |
| help | 2-13 |
| set command assignmentoperator | 2-14 |
| set help | 2-15 |
| show alias | 2-15 |
| show command history | 2-16 |

Introduction

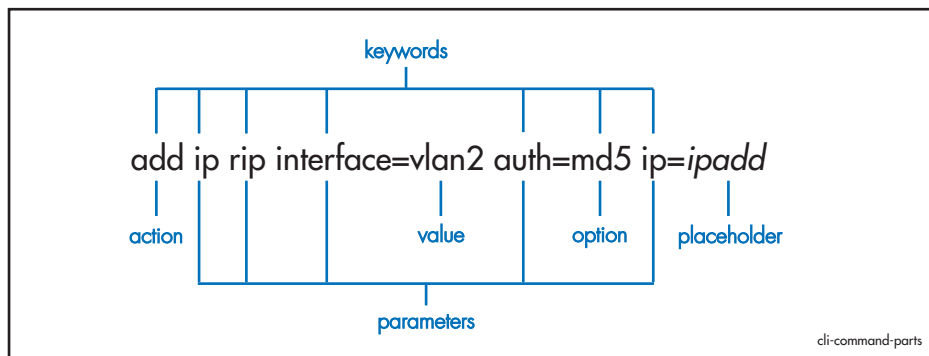
You can set up, manage, monitor, and troubleshoot the switch using the command line interface (CLI) or the web-based GUI. This chapter describes the command line interface (CLI). The CLI provides an extensive set of commands that you can enter from:

- a terminal, or a terminal emulator on a PC, connected to an asynchronous port
- a telnet session for remote access
- a secure shell session for secure remote access

The AlliedWare command structure is not mode-based. This means, for example, that you can configure a feature at any time; you do not have to enter a configuration mode.

Parts of a Command

A command is a sequence of keywords and values that define an action for the switch to perform. This Software Reference uses terms in the following figure and table when describing commands.



| Command Part | Description |
|--------------|---|
| Keyword | A generic term for a predefined sequence of characters that the CLI treats as a single unit. Actions, parameters, and some parameter values are keywords. Keywords are not case sensitive. In this Software Reference and the online help, uppercase letters indicate minimum keyword abbreviations. |
| Action | The first keyword in a command. This defines the type of operation to perform. Actions do not have values. |
| Parameter | Additional keywords that define: <ul style="list-style-type: none"> the object of the action (for example, "ip rip" in the figure above) the details of the action (for example, "auth" in the figure above) Parameters are optional or required, may accept values, and are not case sensitive. Spaces must separate parameters. |

| Command Part | Description |
|--------------|---|
| Value | <p>The value assigned to a parameter. Depending on the parameter, a value can be:</p> <ul style="list-style-type: none"> • an item from a list of option keywords • a number • arbitrary text <p>Values are optional or required. Enter values with the syntax <i>parameter=value</i> or <i>parameter value</i> (for details, see set command assignmentoperator). Most values are not case sensitive, except for text, such as passwords.</p> |
| Option | A keyword that is one of a pre-defined list of values that a parameter can accept. |
| Placeholder | <p>A format convention that describes the value a parameter can accept. Instead of typing the placeholder, replace it with an appropriate value. In this Software Reference, placeholders are printed in lowercase italic font.</p> |
| Default | The value the switch uses as the parameter when you do not enter one but the parameter requires one. |

Entering a Command

Known syntax To enter a command when you know the syntax, simply type the command at the command prompt, then press the Enter key. For example, to create a filter that blocks traffic from the IP address 172.16.248.33, type the following text:

```
add ip filter=1 source=172.16.248.33 action=exclude
```

Command parameters are not case sensitive, although some values are, such as passwords.

Command lines can be up to 1000 characters long, excluding the prompt. The switch assumes that the width of the terminal screen is 80 characters, and wraps lines at the 80th column regardless of the setting of the terminal.

Unknown syntax If you do not know the command syntax, use any of the following to help you:

- the **help** command
- the ? key
- the Tab key

See [“What is Valid Syntax?” on page 2-6](#) and [“What Commands does a Feature Support?” on page 2-8](#).

Shortcuts

Tab completion Press the Tab key to complete actions and parameters after you have typed enough letters to uniquely match the desired action or parameter. The following figure shows an example.

```
Manager > add ospf ra<press Tab immediately afterwards>
Manager > add ospf range (displayed in response)
```

Except when performing Tab completion, the Tab key gives the same results as the ? key (see [“What is Valid Syntax?” on page 2-6](#)).

Tab completion has the following advantages over using [abbreviated syntax](#):

- For many parameters, you must type more characters for the minimum abbreviation than for Tab completion. This is because the abbreviation is often longer than a unique string.
- If you try Tab completion without typing enough letters to make the parameter unique, the switch lists all potential matches. This gives you useful information.

Abbreviated syntax You can abbreviate most keywords; typing just the first few characters is usually adequate.

Abbreviated syntax has the following advantages over using [Tab completion](#):

- You can abbreviate option keywords as well as parameters.
- The command line stays shorter.

Examples

For the **disable ip srcroute** command, type:

```
dis ip srcr
```

For **add ip filter=1 source=172.16.248.33 action=exclude**, type:

```
add ip fil=1 so=172.16.248.33 ac=excl
```

For **add ip interface=vlan1 ipaddress=172.16.248.33 mask=255.255.255.0**, type:

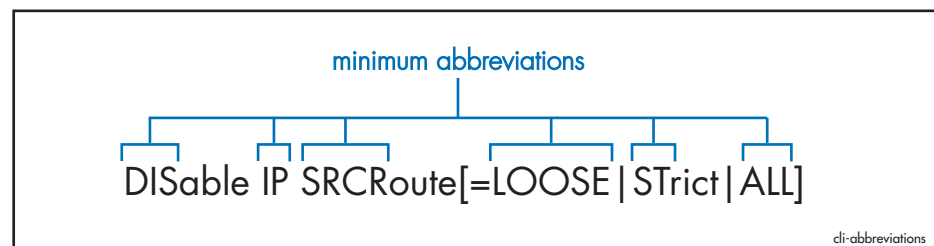
```
add ip int=vlan1 ip=172.16.248.33 mask=255.255.255.0
```

What is the Minimum Abbreviation?

To find out the minimum abbreviation for a keyword, check the command syntax by using one of the following:

- the ? key (see [“What is Valid Syntax?” on page 2-6](#))
- the Tab key (see [“What is Valid Syntax?” on page 2-6](#))
- the [help command on page 2-14](#)
- the Command Reference entry in the Software Reference chapter for that command.

The command syntax shows the minimum abbreviation in uppercase letters, as shown in the following figure.



Aliases An *alias* is a short name assigned to a long character sequence that you use often. An alias may represent part of a command or a complete command.

When the command processor executes a command, it expands all aliases, and then parses and processes the resulting command line normally. Alias expansion occurs when a command is executed from an interactive terminal session or from a script file, except script files run at startup. Alias expansion is not recursive—the command processor scans the command line only once for aliases.

To create a new alias, use the [add alias command on page 2-12](#). To delete an existing alias, use the [delete alias command on page 2-14](#).

To display a list of aliases and their related replacement strings currently defined on the switch, use the [show alias command on page 2-16](#).

Command Sets

Most commands are members of a set of commands that encompass the logical use of a function. The following tables describe the most common sets.

enable, disable

| Parameter | Description | Example |
|-----------|-----------------------|---|
| enable | Turns on a function. | enable ip , which starts the switch processing IP packets. |
| disable | Turns off a function. | disable ip , which stops all IP functionality. |

add, delete, set, show

| Parameter | Description | Example |
|-----------|--|---|
| add | Adds an item to something that already exists, either by default or because you have created it. | add ip route , which adds a route to the route table. The route table exists by default. |
| delete | Removes an item from something. | delete ip route , which removes a route from the route table. |
| set | Changes the values associated with an item. | set ip route , which can change a route's preference. |
| show | Displays the settings of an item, or its contents. | show ip route , which displays the contents of the route table. |

create, destroy, set, show

| Parameter | Description | Example |
|-----------|--|---|
| create | Creates a new item. For many protocols, you must create something before you can add items to it or add it to another item. | create classifier , which creates a new classifier to match certain traffic for further processing. |
| destroy | Removes a previously-created item. | destroy classifier , which removes a classifier from the switch. |
| set | Changes the values associated with an item. | set classifier , which changes the criteria that the classifier uses to identify the relevant traffic. |

| Parameter | Description | Example |
|-----------|-----------------------------------|--|
| show | Displays the settings of an item. | show classifier , which displays the classifiers and what criteria each uses to classify traffic. |

For detailed information about commands for a particular function or protocol, see the Command Reference section in the appropriate chapter of the Software Reference. The Command Reference sections list commands alphabetically.

Chapters also contain background information and procedures for using the function or set of functions.

What is Valid Syntax?

You can use either the Tab key or the ? key to quickly find which parameters you can type next, as summarised in the following table. Note that in earlier software versions, the Tab key recalled the most recent command that matched a partially entered command—use Ctrl+R to do this now.

| To... | Press Tab or ? key after... | Example | Figure |
|--|--|--|--|
| list a one-line description of command actions | the empty command prompt. | ? <Tab> | Figure 2-1 |
| list all parameters that can follow a partial command and, for some parameters, display a one-line description | a partial command, followed by a space after the last letter. | add ? add ospf <Tab> | Figure 2-2 Figure 2-3 |
| list all parameters starting with particular letters, which can follow a partial command | the last letter. Do not type a space after the last letter. If there is only one match, the Tab key completes the parameter. | add ospf r<Tab> add ospf r? | Figure 2-4 |
| list or describe values for a parameter | the parameter and an = sign. Do not type a space after the = sign. | add ospf range=? add ospf range=<Tab> | Figure 2-5 |

Examples The following examples include a ? or <Tab> to show what to type, but the ? and Tab key are not displayed on screen. Note also that some of the parameters in the examples may not exist on your switch.

Figure 2-1: A list and description of all command actions, after typing a question mark

```

Manager >?

ACTivate      Cause an action to be taken immediately
ADD           Add new items to existing objects or instances
CLear         Erase memory (NVS or FLASH) totally - use with extreme caution!
Connect       Connect to a named Telnet or interactive host service or asyn port
COPy          Copy a file in NVS or FLASH memory
CREate        Make a new object or new instance of an object
.
.
.

Manager >

```

Figure 2-2: A list of valid parameters, after typing a question mark

```

Manager > add ?

ACC           Define an ACC call or add an asyn port to the call
ALIAS         Specify a short alternative to a longer command sequence
.
.
.
NTP           Specify the NTP peer. Only one unique peer may be defined
OSPF          Add a host, area, interface, neighbour, range, stub or MD5 key
PIM           Add interface to the PIM list, or configure RP or BRS candidate
.
.
.

Manager > add

```

Figure 2-3: A list of valid parameters, after pressing the Tab key

```

Manager > add ospf <Tab>

INTErface
AREa
HOSt
MD5key
NEIghbour
RANge
REDistribute
STUB
SUMMaryaddre

Manager > add ospf

```

Figure 2-4: A list of matching parameters, after pressing the Tab key

```

Manager > add ospf r<Tab>

RANge
REDistribute

Manager > add ospf r

```

Figure 2-5: A description of the parameter value, after typing a question mark

```

Manager > add ospf range=?

required - an IP address in dotted decimal notation

Manager > add ospf range=

```

What Commands does a Feature Support?

If you are unsure about which command to use, you can list the full syntax of all commands that apply to a particular feature. For example, you can list all DHCP commands.

This is particularly useful if you do not know which command action to use, for example, whether to use an **add**, **create** or **set** command to configure a particular setting. If you already know the beginning of the syntax, it is generally quicker to use the question mark or Tab key. See [“What is Valid Syntax?” on page 2-6](#).

All features To list all supported features, use the command:

help

For example, **help** displays the following output, depending on the switch series.

Help is available on the following topics:

| | |
|-------------------|---|
| HELP ADSL | ADSL commands |
| HELP APPLETALK | Appletalk commands |
| HELP ASYNCHRONOUS | Async ports, TTY, & Asynchronous call control |
| HELP ATM | ATM commands |
| HELP BGP | BGP version 4 routing protocol commands |
| HELP BRIDGE | Bridging commands |
| HELP BOOTP | BootP relay commands |
| HELP CLASSIFIER | Generic packet classifier commands |
| HELP DECNET | DECnet commands |
| HELP DHCP | DHCP server commands |
| . | |
| . | |
| . | |

A specific feature To list commands for a specific feature, use the command:

help feature

For example, **help dhcp** displays the following output.

```
DHCP server commands:
```

```
ENABle DHCP [BOOTp] [DEBug]  
DISABle DHCP [BOOTp] [DEBug]
```

```
CREate DHCP POLIcy=name LEASEtime={lease-time|INFinity} [INHerit=name]  
DESTroy DHCP POLIcy=name
```

```
.  
.  
.
```

Source of help information

The help information comes from a file that is stored in flash memory. Help files have a .hlp filename extension. When you upgrade your product software, also download the associated new help file and then install it by using the command:

```
set help=helpfile
```

To display the current help file, use the command:

```
show system
```

Editing Commands

You can edit commands on the command line by using the keys in the following table.

| To... | Use... |
|---|---------------------|
| move the cursor backwards and forwards within the command line | ← and → |
| move the cursor to the beginning of the command line | Ctrl+A or Home |
| move the cursor to the end of the command line | Ctrl+E or End |
| delete the character to the left of the cursor | Delete or Backspace |
| toggle between insert and overstrike mode | Ctrl+O |
| clear the command line | Ctrl+U |
| interrupt (or "break") text paging or continuously streaming text, for example, when results from show commands are displayed. Buffered text is deleted. | Ctrl+Q |
| terminate the Telnet session before the login is complete | Ctrl+D |

You need not put the cursor at the end of the command line before pressing the Enter key to execute a command.

The default mode for editing is insert mode. In insert mode, characters are inserted at the cursor position and any characters to the right of the cursor are pushed to the right to make room. In overstrike mode, characters are inserted at the cursor position and replace any existing characters.

Recalling Commands

You can recall previously-entered commands and configure the command history by using the keys and commands shown in the following table.

| To... | Use... |
|--|--|
| move backwards through a list of previous commands | ↑ or Ctrl+B |
| move forwards through a list of previous commands | ↓ or Ctrl+F |
| recall the most recent command that matches a partially entered command | Ctrl+R Note that in earlier software versions, the Tab key did this |
| display the command history so that you can select a command from the list | show command history show asyn history |

| To... | Use... |
|--|---|
| clear the command history | <code>reset asyn history</code> |
| specify the maximum number of commands stored in the command history | <code>set asyn history=0..99</code> <code>set tty history=0..99</code> |

Command Reference

This section describes the commands available on the switch to support day-to-day operational and management activities.

The shortest valid command is denoted by capital letters in the Syntax section. See [“Conventions” on page xxxviii of About this Software Reference](#) for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

add alias

Syntax `ADD ALias=name STRing=substitution`

Description This command adds an alias. An *alias* is a short name for a longer version. An alias may represent part of a command or a complete command.

| Parameter | Description |
|-----------|--|
| ALias | The shorter version you would type on the command line. This can be: <ul style="list-style-type: none">• a string 1 to 132 characters long• any printable character• case sensitive If <i>name</i> contains spaces, it must be in double quotes. Default: no default |
| STRing | The longer version that you want to replace the shorter version during processing. This can be: <ul style="list-style-type: none">• a string 1 to 132 characters long• any printable character• case sensitive If <i>substitution</i> contains spaces, it must be in double quotes. Default: no default |

When the command processor executes a command, it expands all aliases, and then parses and processes the resulting command line normally. Alias expansion occurs when a command is executed from an interactive terminal session or from a script file, except script files run at startup. Alias expansion is not recursive—the command processor scans the command line only once for aliases.

When using aliases, we suggest you use the = sign in the syntax to link parameters with their values. Otherwise, if you separate a parameter with a space, a matching alias could erroneously be substituted for the value.

Examples To create the alias “rc” that expands to “show ip route count”, use the command:

```
add ali=rc str="show ip route count"
```

Thereafter, the following commands are equivalent:

```
rc
show ip route count
```

To create an alias “ii” that expands to part of a command (“ip interface”), use the command:

```
add ali=ii str="ip interface"
```

Thereafter, the following commands are equivalent:

```
show ii
```

```
show ip interface
```

Related Commands [delete alias](#)
[set command assignmentoperator](#)
[show alias](#)

delete alias

Syntax `DELeTe ALIas=name`

where *name* is a string 1 to 132 characters long. It may contain any printable character, and is case-sensitive. If *name* contains spaces, it must be in double quotes.

Description This command deletes an existing alias. Occurrences of the alias string in the command line are no longer expanded to the substitution text.

The **alias** parameter specifies the name of the alias to be deleted.

Example To delete an alias with the name "rc", use the command:

```
del ali=rc
```

Related Commands [add alias](#)
[show alias](#)

help

Syntax `HELP [feature]`

Description This command lists the commands that relate to the specified feature. If you do not specify a feature, the switch lists all features for which help is available.

This command accesses a help file. You can change files by using the **set help** command. We recommend that you upgrade your help file when you upgrade the switch software version.

Examples To display help on IP, use the command:

```
help ip
```

Related Commands [set help](#)
[show system](#)

set command assignmentoperator

Syntax SET COMmand {ASSignmentoperator=[Equals|SPaceorequals]}

Description This command sets the assignment operator of the command parser thereby defining the format of the command syntax for the CLI.

| Parameter | Description |
|--------------------|---|
| ASSignmentoperator | Defines the operator between parameters when assigning values. Default: Equals |
| Equals | Requires users to enter = sign. To ensure clarity and accuracy, we recommend always using the = sign. |
| SPaceorequals | Lets users enter either the = sign or just leave a single space between parameters. |

The following commands have the same effect. Note that the first one is clearer because of the = sign.

```
add ip rou=172.16.9.0 mask=255.255.255.0 int=vlan1
next=172.16.8.82 met=1

add ip rou 172.16.9.0 mask 255.255.255.0 int vlan1 next
172.16.8.82 met 1
```

Take care when using aliases because they match any whole word on the command line. Therefore, if you separate a parameter with a space, a matching alias could erroneously be substituted for the value.

Note that certain command handlers, such as STT, PERM, and ACC, always require the = sign.

Examples To set the command processor so that you can enter a space between parameters and values on the command line, use the command:

```
set com ass=sp
```

set help

Syntax SET HELP=*helpfile*

Description This command sets the help file used by the [help command on page 2-14](#), and therefore enables you to change help files. You may need to do this when you upgrade the switch software version, for example.

The **help** parameter specifies the name and location of the HLP file that contains the help information for the switch. If a device is not specified, the default is flash. *Helpfile* is a file name in the format [device:]filename.ext and can be:

- uppercase and lowercase letters
- digits
- the characters ~ ' ! @ # \$ % ^ & () _ - { }

Invalid characters are * + = " | \ [] ; : ? / , < > and wildcards are not allowed.

Examples To set the name of the help file to "88-275a.hlp", use the command:

```
set help=88-275a.hlp
```

Related Commands [help](#)
[show system](#)

show alias

Syntax SHOW ALIAS

Description This command displays all currently defined aliases ([Figure 2-6](#), [Table 2-1](#)).

Figure 2-6: Example output from the **show alias** command

```
Alias ..... rc
String .... show ip route count

Alias ..... ii
String .... ip interface
```

Table 2-1: Parameters in output of the **show alias** command

| Parameter | Meaning |
|-----------|---|
| Alias | Name of the alias. |
| String | String substituted for the alias when it appears in a command line. |

Related Commands [add alias](#)
[delete alias](#)

show command history

Syntax SHow COMmand History

Description This command displays past commands as a menu (Figure 2-7). When you type the command's ID number, the command is displayed at the next prompt. Use the **set asyn** command to define how many commands to save in the history for later recall.

Figure 2-7: Example output from the **show command history** command

```
131 set vrrp 20 portmon off
132 set vrrp 20 portmon on
133 sh vrrp 20
134 sh vrrp 0
135 sh vrrp 21
136 sh vrrp 255
137 sh vrrp none
138 sh vrrp any
139 destroy qos queue2priomap queue 0 bwclass 2 vrrp none
140 destroy qos queue2priomap queue 0 bwclass 2 vrrp any
141 destroy qos queue2priomap queue 0 bwclass 2 vrrp 0
142 destroy qos queue2priomap queue 0 bwclass 2 vrrp 256
143 destroy qos queue2priomap queue 0 bwclass 2 vrrp 17,18
144 destroy qos queue2priomap queue 0 bwclass 2 vrrp 17-19
145 destroy qos queue2priomap queue 0 bwclass 2 vrrp
146 destroy qos queue2priomap queue 0 bwclass 2 vrrp 1
147 destroy qos queue2priomap queue 0 bwclass 2 vrrp 20
148 destroy qos queue2priomap queue 0 bwclass 2 vrrp all

Enter command number>
```

Examples To see a list of past commands, enter:

```
sh com h
```

Related Commands [set asyn](#)
[show asyn](#)

