Chapter Three

ISSUING COMMANDS

Introduction

The Digital Command Language (DCL) is the basic interface tool that allows you to communicate and be productive when working in the OpenVMS operating system environment. You may perform a variety of activities during a session such as:

- Organizing and handling files
- Using an editor to prepare reports, memos, and other documents
- Creating and maintaining program code
- Using electronic mail
- Getting help online

All of these tasks are accomplished by issuing one or more DCL commands.

This lesson introduces the structure and features of DCL.

Objectives

To issue DCL commands, a user should be able to:

- Identify the parts and use them to construct syntactically correct DCL commands;
- Enter DCL commands; and
- Efficiently use DCL features such as command recall, command line editing, and control key sequences.

Building a DCL Command Line

Overview

The Digital Command Language (DCL) uses English-like words for its commands and qualifiers. This use of common words makes it easy to:

- Remember DCL commands
- Enter DCL commands at a terminal
- Recognize and correct syntax errors

DCL commands can be used to:

- Perform file manipulation tasks such as displaying directories or copying, renaming, and deleting files;
- Display information about: the system, other users on the system, devices connected to the system, available resources, and so on; and
- Execute system utilities, applications, or user-written programs.

This section discusses the construction and modification of DCL commands.

Constructing a DCL Command

DCL commands are built somewhat like sentences. There are several parts that go together in a specified order to make a command. Just as in sentences, there are also optional parts that can be added to modify the basic commands.

Elements of a DCL Command

Element	Comment and Example
Verb	The verb of the command line specifies the action of your request. `$ HELP` (In this example, `HELP` is the verb.)
Keyword	Sometimes a verb requires a keyword to further specify the action to be taken. `$ SHOW USERS` (In this example, `USERS` is a keyword for the verb `SHOW`.)
Parameter	The parameter(s) receive the action of the verb. Parameter values might be file specifications, queue names, logical names, or other values. `$ PRINT FILE1.TXT` (In this example, `FILE1.TXT` is a parameter.)
Qualifier	The qualifier(s) of the command line describe or modify the action taken by the verb. Each qualifier is preceded by a slash (/). `$ SHOW PROCESS/ALL` (In this example, `/ALL` is a qualifier.)
Qualifier Value	Some qualifiers accept one or more values. A value assigns a specific quantity to the qualifier. `$ PRINT FILE1.TXT/COPIES=2` (In this example, the number 2 is the value for the qualifier /COPIES.)

Structure Of a DCL Command

The following figure shows the structure of a typical DCL command.

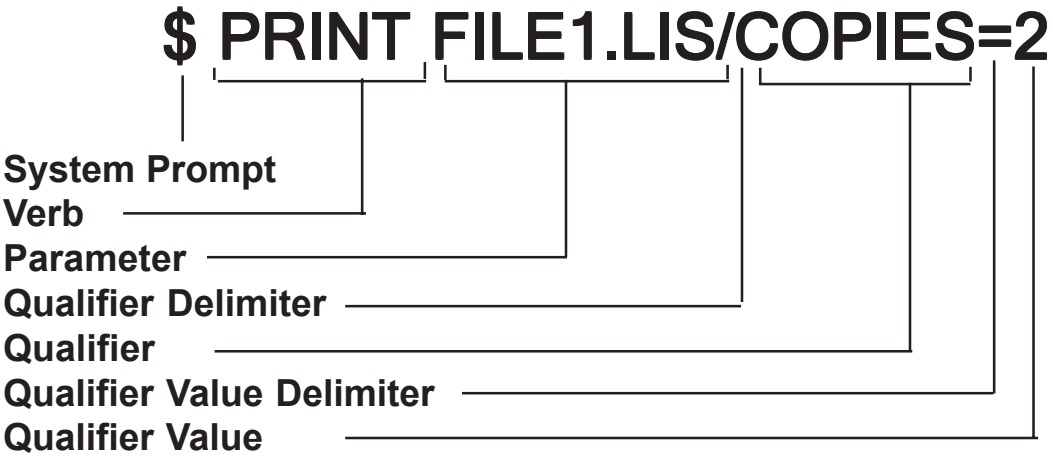

Figure 3-1 Structure of a DCL Command

Notes: Structure of a DCL Command

- The entire line in this figure is called the *DCL command line*.
- **$** is the default OpenVMS system prompt.
- **PRINT** is the *verb* or DCL command.
- **FILE1.LIS** is the *parameter*; it is what is to be printed.
- **/COPIES=2** is composed of a *qualifier* and a *value* which modify the way the PRINT command works. It asks that two copies be printed instead of one.

This command line is terminated by pressing the RETURN key.

Rules For Constructing a DCL Command

Follow these syntax rules when you construct a DCL command:

1. Use any combination of uppercase and lowercase letters. (`LOGOUT` and `logout` are interchangeable.)

2. White space must be included between commands, options, and parameters. One or more spaces or tabs separate these elements; the size of the space does not matter.
   The following two command lines are interpreted as the same by DCL:
   - `$ PRINT FILE1.TXT /COPIES=2`
   - `$ PRINT FILE1.TXT/COPIES=2`

3. Slashes (/) separate qualifiers from commands and parameters. Begin each qualifier with a slash. It does not matter if there are spaces surrounding the slash or not.
   The following command fragments are equivalent:
   - `/COPIES=2`
   - `/ COPIES=2`

4. A comma (,) separates each element in a parameter list, as in the following example:

```
$PRINT FILE1.TXT, FILE2.TXT, FILE3.TXT
```

Modifying DCL Commands

There are two types of qualifiers for DCL commands:

- Command qualifiers
- Positional qualifiers

They are named to designate the parts of the command lines that they modify, either the commands themselves or other information (such as parameters).

Command Qualifiers

Command qualifiers have the same meaning regardless of their position in the command line. For example, the following two commands have the same effect since /HOLD is a command qualifier:

- `$ PRINT/HOLD FILE1.TXT`
- `$ PRINT FILE1.TXT/HOLD`

Both commands place the request in a hold state.

Positional Qualifiers

A positional qualifier may have different meanings depending on its position in the command line.

- When a positional qualifier is placed after the verb, but before the first parameter, the entire command line is affected. The following command requests the printing of two copies of `FILE1.TXT` and two copies of `FILE2.TXT`:

  ```
  $ PRINT/COPIES=2 FILE1.TXT, FILE2.TXT
  ```

- A positional qualifier placed after a parameter affects that parameter only. The following command line requests the printing of one copy of `FILE1.TXT` and two copies of `FILE2.TXT`:

  ```
  $ PRINT FILE1.TXT, FILE2.TXT/COPIES=2
  ```

Using DCL to Interact With The System

Overview

While you are logged in to an OpenVMS system, you conduct a dialogue with the system. You communicate with the system by entering DCL commands. The system responds either by executing your commands or by displaying informational or error messages.

This section discusses:

- Entering DCL commands
- Understanding system messages

Entering DCL Commands

To enter a DCL command:

1. Type the command and any parameters and qualifiers on the terminal keyboard.
2. Press the RETURN key.

Long Command Lines

To facilitate command entry of long DCL commands, you can:

1. Continue typing if your terminal has the wrap feature set.

   Use the SHOW TERMINAL command to see whether your terminal has been set for WRAP or NOWRAP. If WRAP is set, your typing will automatically be continued on the next line.

```
$ PRINT FILE1.TXT/COPIES=3 , FILE2.TXT/COP
IES=2 , FILE3.TXT/COPIES=4
```

Example 3-1 - Long Command Lines

2. Terminate the line with a hyphen (-) and press the RETURN key. The system prompts you with an underscore followed by the prompt, and you can continue typing.

```
$ SHOW -
_$ TIME
```

Example 3-2 - Terminating Lines with a Hypen

The underscore (_) before the prompt (_$) is supplied by DCL when it expects a command line to be continued.

Abbreviating DCL Commands

You can abbreviate any command or qualifier by typing only the first four letters of the command. (Some commands are unique to fewer than four letters, but all DCL commands are guaranteed to be unique to four letters.)

Comment Character

DCL provides the ability to insert comments into a command line. These comments will not be read as part of the command. Use the exclamation point (!) to tell DCL not to read anything beyond that character as part of the command line.

Although there is not much use for the comment character other than in command procedures, you can put a comment on a command line to see how it works. Type the following at your keyboard:

```
$ SHOW USERS !This command will list users on your system
```

Example 3-3 - Using the Comment Exclamation Point

The system will execute the SHOW USERS command and list the users on your system. It will not read the comment after the exclamation point. If you were to enter the same line without the exclamation point, you would get an error message because DCL would try to read the comment as part of the command line.

This is not commonly used during an interactive terminal session, but it can be very useful for documenting DCL command procedures (program files).

Understanding System Messages

The OpenVMS operating system issues messages in response to many of the commands you enter at your terminal. These messages can tell you that a specified operation was not successful and suggest possible causes of error.

The system does not usually display messages that confirm the success of a command. The return of the $ prompt after a command signifies success. However, if you use the /LOG qualifier, the system will return a success message.

Error Messages

Error messages generally occur for the following reasons:

- A command is entered incorrectly. It contains a spelling or syntax error or an incorrect file specification.
- A required resource is currently unavailable. Tape drives, disks, and other devices may not always be available for use.
- A programming error is detected during the execution of an image.
- A system software/hardware problem exists.

Help/Message Utility

Use the Help Message utility (MSGHLP) to get online help regarding system messages. To display information about the status of the last command, type the following:

```
$ HELP/MESSAGE
```

You can also display information about a specific message by including the message identifier or words from the message text. The following example illustrates requesting the help message for the BADACP status text:

```
$ HELP/MESSAGE BADACP
```

The following figure shows a sample of a system error message. The user typed `THINK` as a command.

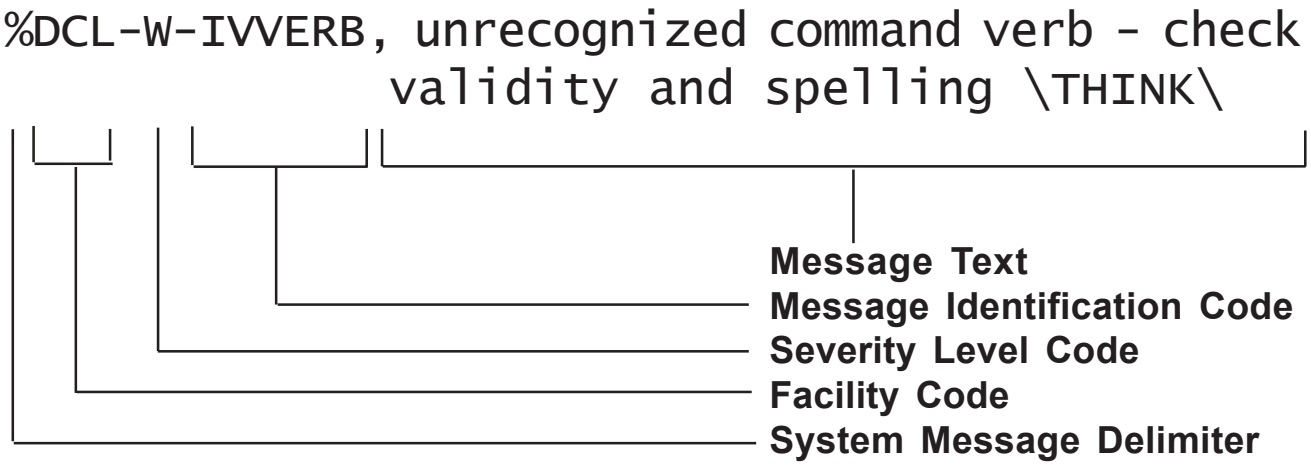

Figure 3-2 Structure of a System Message

Notes: Structure of a System Message

- **%** is the *system message delimiter* and precedes all system messages.
- **DCL** is the *facility code*; it names the portion of the operating system that has detected an error.
- **W** is the *severity level code*; it describes the severity of the error (**W** means warning).
- **IVVERB** is the *message identification code*; it is used to locate information about a message in documentation.
- The *message text* (**unrecognized...**) gives a brief, but more detailed, explanation of the message, and often suggests an action.

The five levels of severity in a system message are:

S (Success)

The command completed successfully.

```
$ COPY/LOG A.TXT B.TXT
%COPY-S-COPIED,  WORK3:[USER]A.TXT;1  copied
to...
```

I (Informational)

The OpenVMS operating system sometimes displays additional information about the success of an operation.

```
$ PURGE/LOG
%PURGE-I-FILPURG, WORK3:[USER]TEST.LIS;2...
%PURGE-I-FILPURG, WORK3:[USER[TEST.LIS;1...
%PURGE-I-TOTAL, 2 files deleted (6 blocks)
```

W (Warning)

Your request may have been performed incompletely.

```
$ DIRECTORY NONEXISTENT.FILE
%DIRECT-W-NOFILES, no files found
```

E (Error)

The output or program result is incorrect.

```
$ DELETE A.TXT
%DELETE-E-DELVER, explicit version number...
```

F (Fatal or Severe)

The operation failed.

```
$ DEASSIGN Y
%SYSTEM-F-NOLOGNAM, no logical name match
```

Using DCL Features

Overview

In addition to supporting an English-like vocabulary and rules of grammar, DCL provides a number of conveniences for entering commands.

This section discusses:

- DCL command line prompting
- Using wildcards on the DCL command line
- Using the control key
- Recalling DCL commands
- Editing command lines

DCL Command Line Prompting

Some DCL commands will prompt you for information if you have forgotten the exact syntax that is expected.

Not every command prompts for parameters, and sometimes a default parameter will be assumed.

For example, perhaps you want to make a copy of a file, but you cannot remember if the old file or the new file comes first in the command line. If you type the COPY command without any parameters, the system will prompt you for the rest of the information.

```
$ COPY
_From: FILE1.TXT
_To: NEW_FILE.TXT
```

Example 3-4 - DCL prompting

Using Wildcards on the DCL Command Line

A ***wildcard*** is a special character that holds single or multiple space(s) in a character string. By using wildcard characters in file specifications, you can apply the action of a DCL command to more than one file at a time.

File Specification Wildcards

The wildcards that are used in file specifications are:

asterisk (*)
Matches any number of legal characters in the position it occupies

percent (%)
Matches any single legal character in the position it occupies, and can occur more than once
- MY%.TEXT looks for MY followed by any single character.
- MY%%.TEXT looks for MY followed by any two legal characters.

Uses For Wildcards

A command entered with wildcards in the file specification will apply to all fields that match the wildcard specification.

```
$ DELETE MY*.TEXT;*
```

The command above deletes all versions of all files that:
- start with MY,
- are followed by any number of legal characters, and
- have a filetype of TEXT.

Caution For Wildcards

The following command should **never** be used:

```
$ DELETE *.*;*
```

The OpenVMS operating system will not ask if you really want to delete all files of all names and types and all versions; it will just delete them.

Popular Qualifiers in DCL Commands

Most DCL commands support one or more qualifiers that affect the way the command is executed. The qualifier can be attached directly to the command, or to one of the parameters in the command line. A number of specific qualifiers are found in many DCL commands and are often used by programmers and developers. The use of these qualifiers allows the user to verify that an operation is desired prior to executing the operation and displays or logs the operation once it is executed. In the case of such commands as PURGE and DELETE, the use of the these qualifiers will save the programmer or developer time and aggravation.

The sections which follow describe some of these more popular qualifiers. Please verify their use on a specific command through online help or the OpenVMS reference manuals.

The /CONFIRM Qualifier

The **/CONFIRM** qualifier controls whether a request is issued before each operation to verify that the operation should be performed. DCL will prompt the user for an answer prior to continuing. Acceptable answers include:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

The answer is not case sensitive nor do you need to enter the entire answer. Answers abbreviated to one or more letters are acceptable. Entering QUIT or pressing Ctrl/Z indicates that you want to stop processing the command. Entering ALL indicates that you want to continue processing without further prompts. The default for this qualifier is **/NOCONFIRM**.

The /LOG Qualifier

The **/LOG** qualifier controls whether the execution of an operation is displayed to the screen or to a log file. The default for this qualifier is **/NOLOG**.

The /PAGE Qualifier

The **/PAGE** qualifier controls whether the display of requested output scrolls until completion or the informations is displayed one screen (20 lines) at a time. The default for this qualifier is **/NOPAGE.**

The Control Key

The control key is marked CTRL on most keyboards and is used in combination with other keys to send special messages to the operating system. You can issue a control sequence by pressing and holding the CTRL key while striking another key, similar to the way that you can capitalize a letter by pressing and holding the SHIFT key while striking the letter. Using control sequences can save you time and keystrokes.

Sample Control Key Sequences

The following examples show how the control key and other keys can be combined to issue special commands.

Key Combination	Function Performed
CTRL/T	Displays a single line of statistical information about the current process.
CTRL/S	Pause information scrolling across the screen. This function allows a user to carefully read a particular section of a long display.
CTRL/Q	Restart the display of information across the screen after it has been stopped with CTRL/S. The display will continue from the point where it was paused.
CTRL/O	Stop and restart (toggle) display of output to the screen. This differs from CTRL/S and CTRL/Q in that it does not pause the output; it stops output from being displayed at all. When you resume output to the terminal screen by issuing another CTRL/O, the output will start at its current point. This is useful in skipping over large amounts of output when you are looking for something in a very large file.

Effect of CTRL/O on Command Execution

Note that if you use CTRL/O, it takes much less time for the computer to process information from a requested command because it does not have to write it out to the screen. For instance, if you are displaying the contents of a very large file, it will take much less time to display several pages while CTRL/O is in effect than it will to display those same pages on the terminal screen.

Recalling DCL Commands

DCL makes it possible to recall previously issued commands in order to modify or re-enter them. This feature can be very useful when:

- a lengthy command must be repeated several times with only slight modification, or
- when a user has made an error in typing a lengthy command string (recalling it eliminates the possibility of making a different error while retyping).

The RECALL Command

The RECALL command displays previously entered commands so that they can be reused. Up to twenty commands are stored in the recall buffer on OpenVMS VAX systems, and up to 254 commands are stored on OpenVMS AXP systems.

Use the RECALL/ALL command to display a listing of the commands in your recall buffer.

```
$ RECALL/ALL
1 SHOW USERS
2 SHOW TERMINAL
3 SET PASSWORD
4 DIRECTORY
$
```

Example 3-5 - Displaying All Commands in the Recall Buffer

Notes: Displaying All Commands in the Recall Buffer

- The RECALL/ALL command displays the previously entered commands in your recall buffer.
- Each command is assigned a number in the display.

The RECALL/PAGE Command

OpenVMS Alpha systems display up to 254 previously entered commands. To make the display of the recall buffer easier to read, the RECALL command has the /PAGE qualifier on OpenVMS AXP systems. This enables you to display the recall buffer one screen at a time.

Recalling and Entering a Command

You can select a specific command to reissue by either of the following:

- Entering the RECALL *n* command, where *n* is the number of the command in the recall buffer.

  $ RECALL 3 would redisplay the third command in the recall buffer.

- Entering the RECALL *command* command.

  $ RECALL SHOW would redisplay the most recent SHOW command.

When you have recalled the desired command, press RETURN and the command will be executed again.

Up and Down Arrow Keys

The system assigns special meaning to certain keys on your keyboard. The up arrow and down arrow keys are two of these special keys. When pressed at the DCL prompt ($), these keys can be used to recall commands, and allow the user to move up and down the list of commands stored in the recall buffer.

Key Combination...	Function Performed:
Up Arrow	Consecutively recalls the last command passed to the DCL command line interpreter.
Down Arrow	Consecutively recalls the next command passed to the DCL command line interpreter.

Other Keys Used To Redisplay Commands

Two control key combinations can be used to redisplay commands.

Key Combination...	Function Performed:
CTRL/B	Consecutively recall the last command passed to the DCL command line interpreter (same as up arrow).
CTRL/R	Redisplay the last unentered command line on your terminal. (On a hardcopy terminal, the system issues a carriage return prior to retyping the current command line).

Editing Command Lines

Entering Commands

Pressing RETURN passes the command line to the DCL command language interpreter (CLI) for execution regardless of the position of the cursor on the line. The RETURN key signals DCL to read all characters on the line and interpret them as a command.

Editing Commands

Since DCL does not translate or execute what you type on the command line until you press the RETURN key, you can make changes as you are typing. This allows you to:

- modify the current command line when you realize a mistake has been made;
- add or delete part of the line; or
- modify a previously issued command which has been recalled from the command buffer.

The same methods can be used to edit a recalled command.

Enabling Command Line Editing

Use the `SET TERMINAL` command to enable command line editing on your terminal:

```
SET TERMINAL/LINE_EDITING
```

Use the **`SHOW TERMINAL`** command to determine whether line editing is enabled.

Two Editing Modes

There are two terminal modes for editing command lines.

Editing Mode:	Way Commands are Edited:
Overstrike	Each character typed replaces the character at the current cursor position. Overstrike is the default mode.
Insert	Characters to the right of the cursor move right to make room for new characters. Nothing is replaced or edited.

Selecting An Editing Mode

Use the `SET TERMINAL` command to set your terminal's editing mode. Enter one of the following commands:

- `SET TERMINAL/OVERSTRIKE`
- `SET TERMINAL/INSERT`

Selecting an editing mode can also be accomplished by toggling the **F14** function key or the **Ctrl/A** key combination.

The **SET/TERMINAL** command is often included as a part of your **LOGIN.COM** file (a command file executed every time you log into an OpenVMS system).

A variety of keys and control key sequences can be used to edit a DCL command line.

Key Combination...	Function Performed:
Left Arrow **Right Arrrow**	Move the cursor to the left or right within the current command line.
CTRL/H or **F12**	Move the cursor to the left end of the command line.
CTRL/E	Move the cursor to the right end of the command line.
CTRL/A or **F14**	Toggle the keyboard between overstrike and insert modes for the current command.
DELETE	Delete the character to the left of the cursor.
LF, **CTRL/J**, or **F13**	Delete the preceding word.
CTRL/U	Delete all characters to the left of the cursor.

It may be convenient to keep a copy of these key sequences near the computer as you are working.

Command Line Editing Using RECALL

1. You mistype a command without noticing it, and press RETURN.

```
$ SNOW DEFAULT
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
\SNOW\
```

2. You enter another command before you notice your error.

```
$ SHOW TIME
14-AUG-1995 13:26.21
```

3. You realize the previous error, and enter RECALL/ALL to list entries in the recall buffer.

```
$ RECALL/ALL
 1 SHOW TIME
 2 SNOW DEFAULT
```

4. You type RECALL 2 to display the incorrectly typed command. The system displays the command line.

```
 $ RECALL 2
 $ SNOW DEFAULT
```

5. Now use control characters to edit the command line. Type LEFT ARROW to position the cursor at the N.

```
 $ SNOW DEFAULT
```

6. Press the H key to overstike the N.

```
$ SHOW DEFAULT
```

7. Press RETURN. Even though the cursor is not at the end of the line, the entire command line is sent to the command line interpreter.

```
$ SHOW DEFAULT
FAL$DISK_USER:[9RB]
```

Using Online Help

Overview

Any user who is logged in to an OpenVMS system can obtain online information about the syntax and usage of any DCL command by invoking the OpenVMS Help facility. Using the Help facility, users can ask very general questions and the facility will prompt them with lists of possible topics. Users can also enter very specific requests that detail exact commands, qualifiers, and parameters.

This section discusses:

- The OpenVMS Help facility
- The Help Message utility (MSGHLP)

The OpenVMS Help Facility

Introduction

The simplest method of invoking the Help facility is to type the DCL HELP command at the OpenVMS prompt:

```
$ HELP
```

Example 3-6 - Issuing the HELP Command

A short description of HELP will then be displayed, as well as a list of available topics. Topics that are all in uppercase letters are DCL commands. General topics are in lowercase letters.

Selecting Topics in HELP

To select a topic from the topic list:

- Enter the `HELP` command;
- Select a topic from the list and type it after the `Topic?` prompt; and
- Select a subtopic and type it after the `Subtopic?` prompt.

You can type INSTRUCTIONS for more detailed instructions on how to use HELP. Also, you can type HINTS if you are not sure of the name of the command or topic for which you need help.

You can abbreviate any topic name, although ambiguous abbreviations result in a display of all possible matches.

To return to the original list of topics or a previous list of subtopics, use the following rules:

- Press the Return key to list the topics or subtopics on the previous level.
- Type a question mark (?) to redisplay the most recently requested text.

Ending A Help Session

You can end your help session by either of the following methods:

- Press the Ctrl/Z key to exit from the Help facility. This key combination echoes on your screen as Exit;
- Press the Return key one or more times until the DCL $ prompt is displayed.

You can enter the commands in the following examples at your terminal as you read through them. Some of the output in the examples have been truncated for space reasons; you will see the complete displays on your terminal screen.

```
$ HELP
HELP
The HELP command invokes the HELP Facility to display information
about a command or topic.  In response to the "Topic?" prompt, you
can:
     •
     •
     •
Additional information available:
:=   =    @          ACCOUNTING     ALLOCATE  ANALYZE   APPEND
CALL CANCEL    CC    CLOSE          CONNECT   CONTINUE  CONVERT
     •
     •
```

Example 3-7 - Entering The HELP Command

```
Topic?  SHOW

SHOW
     Displays information about the current status of a process,
     the system or devices in the system.

     Format

          SHOW option

Additional information available:
Description    ACCOUNTING     AUDIT     BROADCAST CLUSTER
     •
     •
     •
SHOW Subtopic?  SYSTEM

SHOW
     SYSTEM
          Displays status information about current processes.

          Format

               SHOW SYSTEM

     Additional information available:

     Qualifiers
     /BATCH         /CLUSTER  /FULL     /EXACT    /HEADING
     /HIGHLIGHT     /IDENT
     •
     •
     •
```

Example 3-8 - Accessing HELP Subtopics

This is an example of what the display will look like when you press the Ctrl/Z key to exit from the HELP utility.

```
SHOW Subtopic?  Exit
$
```

Example 3-9 - Exiting the HELP Command

Selecting a Topic From The DCL Command Line

In addition to selecting a topic from those available within the HELP environment, you can select one from the DCL command line by typing the topic immediately after the HELP command.

```
$ HELP SHOW
```

Example 3-10 - Issuing HELP With a Topic

Then you can specify further information on the command line.

```
$ HELP SHOW USERS
```

Example 3-11 - Issuing HELP With a More Specific Topic

The more detailed the information you enter following the HELP command, the more specific will be the response to your request. For example, the following HELP command will result in more detail than the previous two examples.

```
$ HELP SHOW USERS /FULL
```

Example 3-12 - Issuing HELP With a Topic and Qualifier

To suppress the HELP facility's introductory display, use the /NOINSTRUCTION qualifier.

```
$ HELP /NOINSTRUCTION
```

Example 3-13 - Issuing HELP With the NOINSTRUCTION Qualifier

Help Message Utility (MSGHLP)

The HELP/MESSAGE Command

Use the Help Message Utility (MSGHLP) to get online help for system messages. To display information with regard to the last command, enter the following:

```
$ HELP/MESSAGE
```

Example 3-14 - Issuing the HELP/MESSAGE Command

You can also display information about a specific message by including the message identifier or words from the message text. For example:

```
$ HELP/MESSAGE BADACP
```

Example 3-15 - Issuing HELP/MESSAGE With a Topic

Using Status Codes

A message and its description can also be accessed by entering the message status code. For example:

```
$ HELP/MESSAGE/STATUS=%X00038090
```

Example 3-16 - Issuing HELP/MESSAGE With the STATUS Qualifier

If you do not know the message status code, you can view it by entering the command SHOW SYMBOL, followed by the $STATUS global symbol. For example:

```
$ SHOW SYMBOL $STATUS
     $STATUS == "%X00038090"
```

Example 3-17 - Using SHOW SYMBOL To Get a Status Code

Summary

Concepts

Constructing a DCL Command

- The construction of DCL commands is similar to that of English sentences.
- Commands and positional qualifiers are used to modify the actions of DCL commands.

Using DCL To Interact With The System

- DCL offers a number of helpful features, including command continuation and abbreviation.
- OpenVMS system messages tell you whether or not a specified operation was successful, and suggest possible causes of error.

Using DCL Features

- DCL will prompt the user for missing parameters.
- Use the asterisk (*) and percent (%) wildcards to apply the action of a single DCL command to multiple files.
- The control key (CTRL) modifies the meaning of many terminal keys.
- Use the RECALL command or the up arrow key to recall previously issued commands, either to modify or re-enter them.
- You can edit the DCL command line using a variety of terminal keys.

Using Online Help

- You can obtain online information about the syntax and usage of any DCL command by invoking the OpenVMS Help facility.
- You can obtain online information about system messages by using the Help Message utility (MSGHLP).

COMMANDS

Using DCL Features

- **RECALL**
  Displays the most recently entered command.
- **RECALL** ***n***
  Displays the *nth* command stored in the recall buffer.
- **RECALL** `command`
  Displays the most recent iteration of the specified command in the recall buffer.
- **RECALL/ALL**
  Displays all commands stored in the recall buffer.
- **RECALL/PAGE**
  Displays all commands stored in the recall buffer, one screen at a time (OpenVMS Alpha systems only).

Using Online Help

- **HELP**
  Invokes the OpenVMS Help facility to obtain online help.
- **HELP/MESSAGE**
  Invokes the Help Message utility to obtain online help about system messages.