

Command Language

User's Guide

digital™

Third Edition

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright © 1985 by Digital Equipment Corporation
All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	Rainbow
DEC	PDP	RSTS
DECmate	P/OS	RSX
DECsystem-10	PRO/BASIC	Tool Kit
DECSYSTEM-20	PRO/Communications	UNIBUS
DECUS	Professional	VAX
DECwriter	PRO/FMS	VMS
DIBOL	PRO/RMS	VT
digital ™	PROSE	Work Processor
	PROSE PLUS	

Before You Start

WHO SHOULD READ THIS BOOK

This book explains how to use the DIGITAL Command Language, a software application that provides an interface between you and the Professional Operating System (P/OS).

CONVENTIONS USED IN THIS BOOK

This book uses certain conventions to represent the way an action should be performed or the way a command should be entered from the keyboard. The following is an explanation of these conventions.

KEY	Words set in this typeface indicate a key on your LK201 keyboard. For example, RETURN, K, and 5 are keys.
CTRL/	Indicates that you hold down the CTRL key while you press a second key. For example, CTRL/Z means hold down the CTRL key while pressing the Z key.
< >	Angle brackets are used in examples to indicate a single keystroke. For example, <EXIT> tells you to press the EXIT key.

[]	Square brackets enclose items that are optional in a command. You do not type the brackets when you type a command. There is one important exception: square brackets in directory names must be included with the directory names. (See the section in this chapter on File Specifications for information about directory names.)
—	An underscore is used in a command item that is composed of two words to make the name more readable, for example, /TASK_NAME. You can type the underscore with the command item if you want, but you do not have to.
...	An ellipsis indicates that you can enter more than one of the items named just before the ellipsis. You separate items with a comma. You do not type an ellipsis in your commands.
Press	Tells you to make one keystroke with the indicated key.
Type	Tells you to enter characters from the keyboard. For example, if you are asked to type DIRECTORY, you make nine keystrokes: one for each letter.
Choose	Implies that you select an item from a list of menu options, and then press the DO key.
<i>Italics</i>	Words defined in the glossary appear in italics the first time they are used in the book.
UPPERCASE	When uppercase is used in format descriptions, it means you have to type exactly what is written. This is called a keyword. For example, TODAY is a keyword.
lowercase	In format descriptions, lowercase words are the names of the items you enter, not the actual items. For example, filespec means you must enter an actual file specification.

Bold

In the detailed descriptions of commands in Chapter 3, the shortest abbreviations you can use are shown in boldface. For example,

\$ DELETE <DO>

Dot matrix

Indicates text that is displayed on the computer screen.

Red ink

Indicates user input in examples.

DEASSIGN

Deletes a logical name assignment previously specified using the ASSIGN or DEFINE command.

DEFINE

Associates a logical name with a physical device, file specification, or another logical name.

DELETE

Removes one or more files from a particular device (disk or diskette volume).

DELETE/DIRECTORY

Removes an empty directory.

DIBOL

Invokes the PRO/Tool Kit DIBOL compiler to compile DIBOL source code files.

DIRECTORY

Displays information about one or more files in a particular directory on a particular volume.

DISMOUNT

Declares a volume (and therefore the files and data it contains) removed from the system. The files and data on the volume are no longer available for use.

EDIT or EDIT/PROSE

On hard disk systems only, runs the PROSE editor, which allows you to create and modify text files.

EDIT/EDT

Runs an interactive editing session with the PRO/EDT editor to create or modify a file on hard disk systems only.

EXIT

Discontinues the running of DCL. On hard disk systems, EXIT results in a display of the menu on which DCL was installed.

FIX

Loads and locks a task into its partition.

FORMAT

Sets up a hard disk so that it is recognized by its controller.

FORTRAN

Invokes the PRO/Tool Kit FORTRAN-77 compiler to compile FORTRAN language source files.

HELP

On hard disk systems only, displays information about most of the available DCL commands. On diskette systems, HELP is limited to a short description of DCL commands.

INITIALIZE

Deletes all existing files on a volume, imparts the standard FILES-11 file structure.

INSTALL

Places a program into the system so that the program can be run. (This is not the same as installing an application described in the *Hard Disk System User's Guide*.) A program is also called a task. The system remembers where the program resides, so the program can be run by the task name rather than its file specification.

Contents

CHAPTER 1 INTRODUCTION

Installing DCL on the Hard Disk System	4
Running DCL on Diskette Systems	4
Overview of DCL Commands	5

CHAPTER 2 USING DCL

How to Enter a DCL Command	13
How to Correct a Mistake Without Retyping the Entire Command	14
How to Display Helpful Information	14
How to Exit DCL	14
DCL Command Format	14
Command Name	14
Qualifiers	15
Parameters	15
Arguments	15
Prompts	16
How to Terminate a Command Without Executing It	17
Abbreviations	17
File Specifications	18
Wildcards in File Specifications	19
File Protection	21

Specifying Files for Network Operations	24
Node Specification	24
Examples	25
Foreign Files	25
Indirect Command Files	26

CHAPTER 3 DCL COMMANDS

ABORT	32
APPEND	33
ASSIGN	37
ASSIGN/TASK	41
BASIC	42
CANCEL	43
CLEAR	44
COBOL	45
CONTINUE	52
COPY	53
CREATE	60
CREATE/DIRECTORY	63
DEASSIGN	65
DEFINE	67
DELETE	71
DELETE/DIRECTORY	74
DIBOL	75
DIRECTORY	78
DISMOUNT	83
EDIT or EDIT/PROSE	84
EDIT/EDT	86
EXIT	89
FIX	90
FORMAT	92
FORTRAN	93
HELP	99
INITIALIZE	101
INSTALL	102
MAIL	104
MOUNT	105
PASCAL	106
PHONE	109
PRINT	110
PRINT/REMOTE	111

PURGE	112
REMOVE	115
RENAME	116
RUN installed task	119
RUN uninstalled task	122
SET [DAY]TIME	124
SET DEFAULT	125
SET DEVICE	126
SET HOST	128
SET PRIORITY	129
SET PROTECTION	130
SET PROTECTION/DEFAULT	134
SET PROTECTION/NODEFAULT	136
SET TERMINAL	137
SHOW CLOCK_QUEUE	142
SHOW COMMON	143
SHOW [DAY]TIME	145
SHOW DEFAULT	146
SHOW DEVICES	147
SHOW LOGICALS or SHOW ASSIGNMENTS	148
SHOW PROTECTION	149
SHOW TASKS/ACTIVE	150
SHOW TASKS/INSTALLED	154
SHOW TASK/LOGICAL_UNITS	156
SHOW TERMINAL	157
SPAWN	159
START	160
START/UNBLOCK	161
STOP/BLOCK	162
SUBMIT/REMOTE	163
TYPE	164
UNLOCK	167

APPENDIX A COMMON ERROR MESSAGES

Common Error Messages	171
-----------------------------	-----

APPENDIX B INSTALLING THE PRO/EDT EDITOR

Installing the PRO/EDT Editor	189
-------------------------------------	-----

APPENDIX C DEVICE NAMES

Local Diskettes	193
Hard Disks	194
The User's Personal Area	194
Special Devices	195
Using LB:	196

INDEX

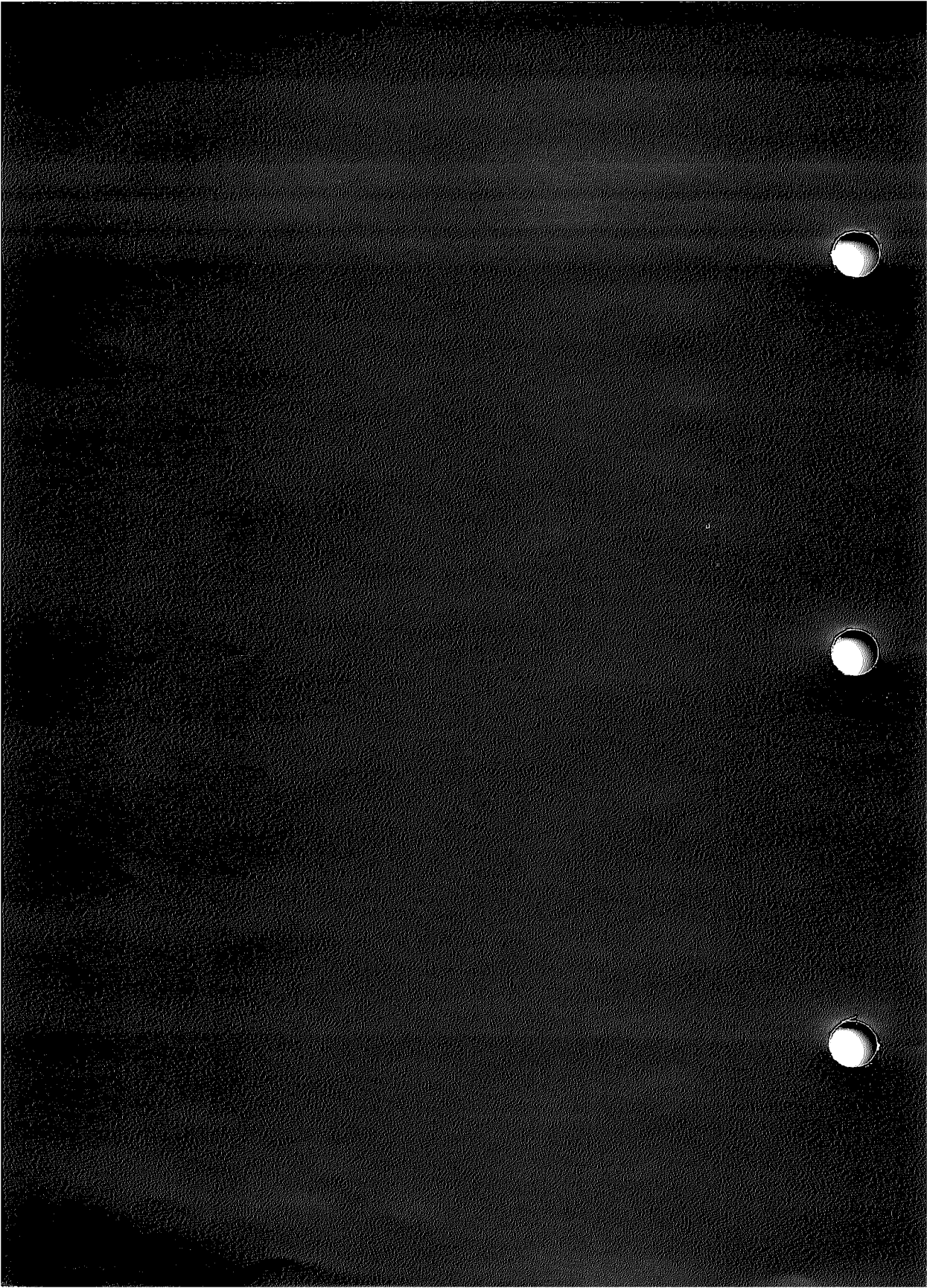
Index	199
-------------	-----

TABLES

2-1 Directory Format Table	26
2-2 File Specification Format Table	26
2-3 Examples of Foreign File Formats	26
3-1 Arguments for the /SHOW Qualifier	77
3-2 Task Status Flags	152
C-1 Device Names and Their Translations	196

1

Introduction



Chapter 1

Introduction

This chapter introduces the DIGITAL Command Language (DCL), an application provided with your Professional Operating System (P/OS) software.

If you are not familiar with the Professional, you should first read your *Hard Disk System User's Guide*.

NOTE: The PRO/Tool Kit and end-user versions of DCL have been merged so that only one copy of DCL exists. If you get the following error message,

```
Sorry, task not installed
```

generally, it means that the command you have tried to use is Tool Kit specific and you do not have the Tool Kit installed.

The DIGITAL Command Language is an interface between you and the Professional Operating System (P/OS). DCL is an efficient way of giving instructions to the Professional.

DCL allows you to manipulate files, run programs, and control and display some system functions. With versions of P/OS later than Version 2.0A, DCL allows file manipulation across DECnet networks for Professionals that have the PRO/DECnet software and hardware installed for the Hard Disk System.

DCL is provided primarily for people who have had previous experience using DIGITAL operating systems. Parts of this book use some terms that you may not understand unless you have had previous experience with RSX-11M/M-PLUS or other operating systems.

The DIGITAL Command Language uses commands that are complete English words. Each command (word) describes the action to perform. For example, if you want to display the date and time, type SHOW DAYTIME.

You type DCL commands in a particular format, described in Chapter 2 of this manual. Chapter 2 also explains how to use DCL and gives an overview of P/OS file specifications. Chapter 3 describes the DCL commands in detail.

INSTALLING DCL ON THE HARD DISK SYSTEM

Before you can use DCL on the hard disk system, you must install it using P/OS Install Application Services. The diskette labeled P/OS HARD DISK COMMAND LANGUAGE and P/OS HARD DISK COMMAND LANGUAGE HELP contain the DCL application.

To install DCL, the system manager must first copy the Command Language application to the public library. Once the system manager has done this, each user can then install the application on his or her menu by selecting "Install Application" from the Environment Menu. A submenu will appear. Choose "Library" from the submenu. Each user can then select the Command Language application from the application group he or she installed it in. The *Hard Disk System User's Guide* describes the installation procedure in more detail.

RUNNING DCL ON DISKETTE SYSTEMS

To run the Command Language application on P/OS Diskette System, insert the diskette labeled P/OS DISKETTE COMMAND LANGUAGE when the system prompts you to insert an application diskette, and press the RESUME key.

OVERVIEW OF DCL COMMANDS

The following commands are available with this DCL:

ABORT	DIBOL ^{1,3}	PRINT
APPEND	DIRECTORY	PRINT/REMOTE ²
ASSIGN	DISMOUNT	PURGE
ASSIGN/TASK	EDIT or EDIT/PROSE	REMOVE
BASIC ^{1,3}	EDIT/EDT ¹	RENAME
CANCEL	EXIT	RUN installed task
CLEAR	FIX	RUN uninstalled task
COBOL ^{1,3}	FORMAT	SET
CONTINUE	FORTRAN ^{1,3}	SHOW
COPY	HELP	SPAWN
CREATE	INITIALIZE	START
CREATE/DIRECTORY	INSTALL	START/UNBLOCK
DEASSIGN	MAIL ^{1,2}	STOP/BLOCK
DEFINE	MOUNT	SUBMIT/REMOTE ²
DELETE	PASCAL ^{1,3}	TYPE
DELETE/DIRECTORY	PHONE ^{1,2}	UNLOCK

¹ Not available on the diskette system.

² Only available if you also have PRO/DECnet on the hard disk system with the extensions to DCL installed. These extensions also allow you to use MAIL and SET HOST commands. Refer to the *PRO/DECnet User's Guide* for details on MAIL and SET HOST and on installing the DCL extensions.

³ Only available if the PRO/Tool Kit compiler for these languages is installed.

The list below briefly describes the commands. Chapter 3 presents a detailed description of the commands, in alphabetical order.

ABORT	Forces an immediate end to a task that is running.
APPEND	Places one or more files at the end of a specified file. For example, you can use the APPEND command to add text to the end of another text file.
ASSIGN	Associates a logical name with a physical device, file specification, or another logical name. This is a way to create an abbreviation that may be easier to use when referencing devices and/or files.
ASSIGN/TASK	Reassigns the logical unit numbers (LUNs) of an installed task from one physical device to another.

BASIC	Invokes the PRO/Tool Kit BASIC-PLUS-2 compiler to begin a BASIC session.
CANCEL	Eliminates entries from the clock queue.
CLEAR	Clears the screen, and optionally, performs a reset-to-initial state.
COBOL	Invokes the PRO/Tool Kit COBOL-81 compiler to compile COBOL language source files.
CONTINUE	Resumes the execution of a task that has been previously suspended.
COPY	Duplicates the contents of one or more files. COPY allows you to copy files from one location to another, with the option to change the names of the destination file(s).
CREATE	Creates a sequential file and enables you to type text directly into the file from your terminal. CREATE is also used for certain network operations with PRO/DECnet to create files.
CREATE/DIRECTORY	Creates a new directory on a specified volume.
DEASSIGN	Deletes a logical name assignment previously specified using the ASSIGN or DEFINE command.
DEFINE	Associates a logical name with a physical device, file specification, or another logical name.
DELETE	Removes one or more files from a particular device (disk or diskette volume).
DELETE/DIRECTORY	Removes an empty directory.
DIBOL	Invokes the PRO/Tool Kit DIBOL compiler to compile DIBOL source code files.
DIRECTORY	Displays information about one or more files in a particular directory on a particular volume.
DISMOUNT	Declares a volume (and therefore the files and data it contains) removed from the system. The files and data on the volume are no longer available for use.
EDIT or EDIT/PROSE	On hard disk systems only, runs the PROSE editor, which allows you to create and modify text files.

EDIT/EDT	Runs an interactive editing session with the PRO/EDT editor to create or modify a file on hard disk systems only.
EXIT	Discontinues the running of DCL. On hard disk systems, EXIT results in a display of the menu on which DCL was installed.
FIX	Loads and locks a task into its partition.
FORMAT	Sets up a hard disk so that it is recognized by its controller.
FORTTRAN	Invokes the PRO/Tool Kit FORTRAN-77 compiler to compile FORTRAN language source files.
HELP	On hard disk systems only, displays information about most of the available DCL commands. On diskette systems, HELP is limited to a short description of DCL commands.
INITIALIZE	Deletes all existing files on a volume, imparts the standard FILES-11 file structure.
INSTALL	Places a program into the system so that the program can be run. (This is not the same as installing an application described in the <i>Hard Disk System User's Guide</i> .) A program is also called a task. The system remembers where the program resides, so the program can be run by the task name rather than its file specification.
MAIL	Invokes the Mail Utility for PRO/DECnet. This command is used with PRO/DECnet on hard disk systems only.
MOUNT	Declares a volume (and therefore the files or data it contains) to be logically known to the system, and available for use.
PASCAL	Invokes the PRO/Tool Kit PASCAL compiler to compile one or more PASCAL source programs.
PHONE	Invokes the Phone Utility for PRO/DECnet. This command is used with PRO/DECnet on hard disk systems only.
PRINT	Used to print files.

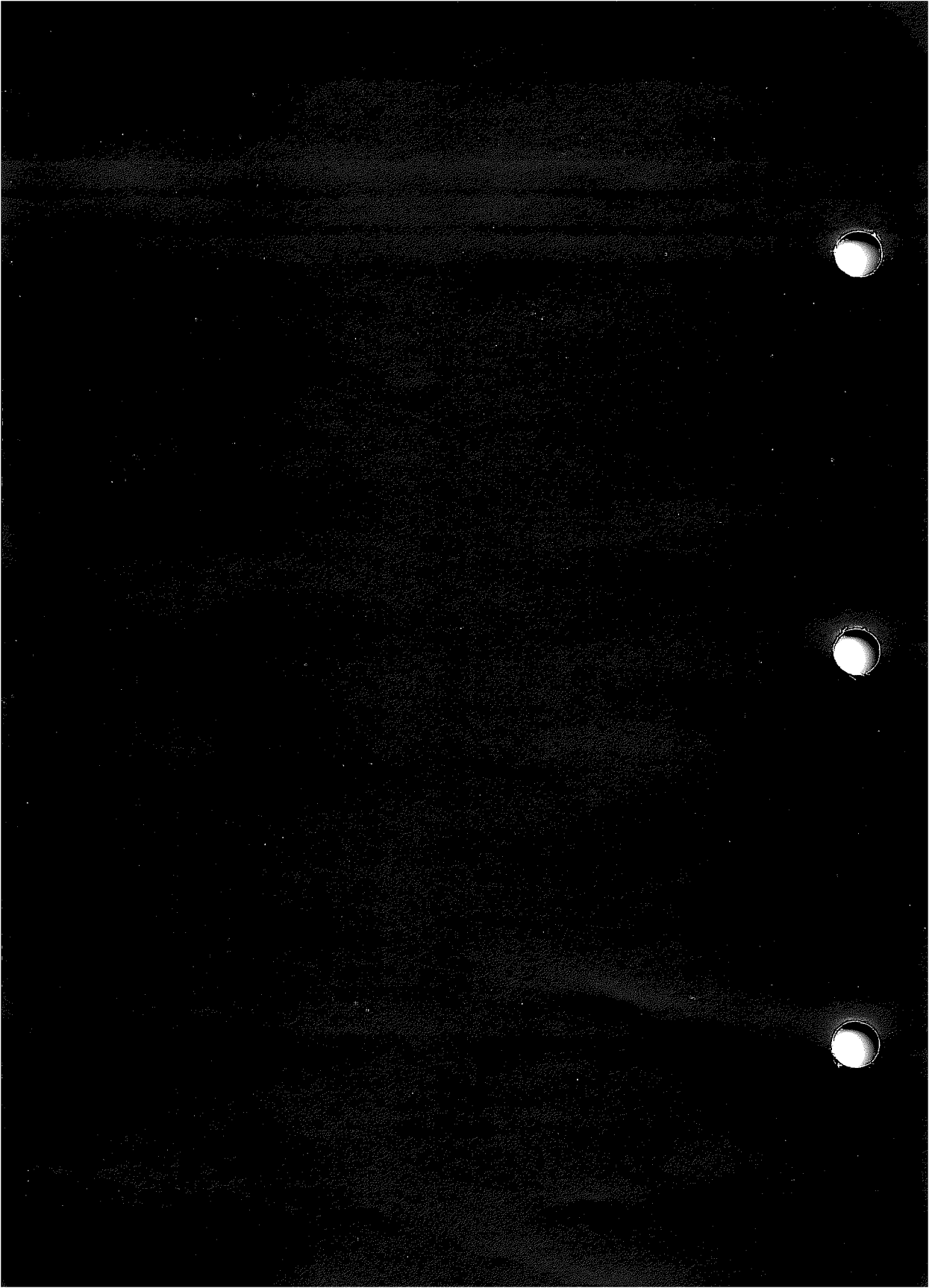
PRINT/REMOTE	Available only with PRO/DECnet. This command directs an existing file on a remote node to a line printer for printing.
PURGE	Deletes all versions of specified file(s) except the most recent version.
REMOVE	Deletes a task (program) name from the system's list of installed tasks, so that the task is no longer installed.
RENAME	Allows you to change the name of a file.
RUN	Takes a compiled and linked program (task), and tells the computer to execute it.
SET	Allows you to define system characteristics such as the current date and time, the current directory, and file protection characteristics.
SHOW	Allows you to display the same information you can set using the SET command.
SPAWN	Allows tasks to be executed in the background.
START	Resumes the execution of a task that was halted by a STOP\$\$ directive.
START/UNBLOCK	Continues the execution of a task that was blocked by the STOP/BLOCK command.
STOP/BLOCK	Blocks an installed, active task.
SUBMIT	Used with /REMOTE for initiating the execution of a command file (see @FILENAME, which follows) at a remote node. SUBMIT requires PRO/DECnet.
TYPE	Displays the contents of one or more files on your screen.
UNLOCK	Attempts to make available one or more files that have become locked by the system due to a defect in an application, a power failure while an application is in progress, or pressing INTERRUPT DO while an application is in progress. (It is not always possible to unlock a locked file, as important information about the file may have been destroyed.)

@FILENAME

Where filename is the name of an indirect command file, instructs DCL to execute a list of DCL commands stored in a command file. This is a shortcut to typing each of the commands every time you wish to use them as a group. See Chapter 2, Indirect Command Files.

2

Using DCL



Chapter 2

Using DCL

HOW TO ENTER A DCL COMMAND

After selecting “DCL” from the Main Menu (or Additional Applications, if you installed your DCL there), a \$ prompt appears at the top of the screen. You enter a command by typing it after the \$ prompt. Then press either DO or RETURN.

For example:

```
$ SHOW DEFAULT <DO>
```

In this example, P/OS will respond by displaying your default device and/or directory.

If an error is made, an error message will appear and you can retry. If you find an error message difficult to understand, refer to the section of Chapter 3 that describes the command you are trying to use. Make sure you have typed everything correctly in the format described. Appendix A contains further information about error messages.

You must enter each DCL command according to a specific set of rules, called the command format. Sections in this chapter describe the DCL command format in detail.

HOW TO CORRECT A MISTAKE WITHOUT RETYPING THE ENTIRE COMMAND

If you make a mistake before you press DO, you can correct it without retyping the entire command by pressing the LEFT ARROW key until the cursor is just to the right of any character(s) you want to change. Delete the part you do not want with the `<X>` key and retype. To add a part you left out, move the cursor to that place and type in the omitted part. The rest of the command will move to the right to make room.

If you make a mistake after you have pressed DO, press the UP ARROW key. The command will be redisplayed for changing. You can display up to nine previous commands this way. You can then use the DOWN ARROW key to display the next command.

While entering DCL commands, the CANCEL key erases the current command so you can start over. The F13 function key erases from where the cursor is to the end of the line.

HOW TO DISPLAY HELPFUL INFORMATION

To display information about DCL commands, either press the HELP key or type HELP and press DO after the \$ prompt. Follow the instructions on the screen.

HOW TO EXIT DCL

To leave DCL, press the EXIT key, or type EXIT and press DO after the \$ prompt. For example:

```
$ EXIT <DO>
```

DCL COMMAND FORMAT

The format for a DCL command is:

```
command-name[/qualifier[,...]] parameter[/qualifier[,...]]
```

Command Name

A command name is usually a verb. It describes the action P/OS is to take. For example:

```
$ HELP <DO>
```

This command will cause help information to be displayed on your screen.

Qualifiers

A qualifier modifies the action of the command. Qualifiers are usually optional, always preceded by a slash (/) and used to perform a more specific action. For example:

```
$ TYPE/TODAY SCHED.DOC <DO>
```

In this example, the qualifier /TODAY modifies the TYPE command so that it displays the latest version of a file named SCHED.DOC created on the current date.

Some commands use qualifiers while other commands use parameters (see next section). You can leave a space between the command or parameter and the qualifier. You *must* leave a space between HELP and any of its qualifiers.

Some of the qualifiers that work locally do not work over DECnet networks. For example, if you try to use the time qualifiers such as /SINCE, /THROUGH, or /TODAY with commands that affect files at other nodes, a syntax error message will result. If this happens, retry the command without those qualifiers.

Some of the PRO/DECnet qualifiers must be specified after the node is specified in the command line or an error message will result.

Parameters

Parameters describe the object of the command. The object may be a specific file to be manipulated, or a keyword. Parameters are preceded by a space. Some commands do not require parameters, while others do. Here is an example of a parameter used in a command:

```
$ TYPE MYFILE.TXT <DO>
```

In this example, the parameter MYFILE.TXT is a text file to be displayed.

Arguments

Parameters may take arguments. Arguments are preceded by a colon (:). For example:

```
$ TYPE/TODAY *.TXT/EXCLUDE:MYFILE.TXT <DO>
```

In this example, the qualifier **EXCLUDE** takes the argument **MYFILE.TXT**. The command requests a display of all files of the type **.TXT** edited on the current date, except **MYFILE.TXT**.

An equal sign (=) can be used instead of a colon, if you prefer. For example:

```
$ TYPE/DATE = 15-FEB-85 *.TXT <DO>
```

In this example, a display of all files of the type **.TXT** created on February 15, 1985 is requested.

NOTE: A colon in a device or volume name *cannot* be replaced with an equal sign. Refer to the section on File Specifications for information about device and volume names.

PROMPTS

Besides the initial \$ prompt, DCL can display other prompts. For example, if you type **RENAME** and press **DO** without typing in a file name, you will be prompted for it:

```
$ RENAME <DO>
Old File Name?
```

In this example, you are prompted to type the name of the file to be renamed. You must supply an answer to a prompt. If you are uncertain what to do, type a question mark (?). For example:

```
$ TYPE <DO>
File(s)? ?
```

Entering a ? provides the following information.

```
TYPE
```

Displays the contents of the designated text files on your screen.

```
Format   ->   TYPE[/qualifier] file-spec[,...]
Example  ->   TYPE PRDCT.DOC,MEM003.TXT,DESIGN.BAS
           Displays the three files PRDCT.DOC,
           MEM003.TXT, and DESIGN.BAS.
```

Additional information available:

```
Qualifiers
/DATE      /EXCLUDE      /SINCE      /THROUGH      /TODAY
File(s)?
```

In this example, information about the **TYPE** command is displayed, and the **File(s)?** prompt reappears for you to enter the qualifiers and parameters you want.

HOW TO TERMINATE A COMMAND WITHOUT EXECUTING IT

To stop the execution of a command that is in the process of doing something, hold down the **CTRL** key and type **C** (referred to as **CTRL/C**).

To exit to the DCL **\$** prompt while another DCL prompt is displayed, use **CTRL/Z**.

ABBREVIATIONS

All DCL commands and qualifiers can be abbreviated to five characters. Many can be abbreviated even further.

In Chapter 3, the shortest usable command abbreviation is indicated by bold letters in each command format description.

The **HELP BRIEF** command displays some commonly used abbreviations. If your command is too long to fit on one line, type a hyphen at the end of the first line and press **RETURN**. The DCL continuation prompt (**\$-**) appears, indicating that you can continue entering the command line. You cannot type a command line longer than 250 characters. Here is an example of line continuation:

```
$ PURGE ACCOUNTS.DAT,CHECKBOOK.BAS,MACDONALD.DOC,- <DO>
$-SPENCER.DOC,LFIELD.TXT,PAYROLL.DAT,BILLS.DAT <DO>
```

If you omit the hyphen at the end of a line, an error message or a prompt for more input will be displayed when you press **DO** or **RETURN**. If an error message appears, you can then press the **UP ARROW** key and correct your mistake by inserting the hyphen at the end of the line.

Another method of continuing a long command line is to use the DCL prompts. For example, when using the **COPY** command, type **FILE** and press **DO**. DCL will then prompt you for the file specification(s) of the file(s) to copy. Type them in and press **DO** again. DCL will prompt you for the file to copy to. This gives you more space to type a long list of files to be copied.

FILE SPECIFICATIONS

A file specification provides the computer system with all the information it needs to identify a unique file. General information about P/OS files is contained in the Introduction to the P/OS File System in the *Hard Disk System User's Guide*. Additional information about RMS-11 files is contained in *PRO/RMS-11: An Introduction* (Order No. AA-P098A-TK). RMS (Record Management Services) is one of DIGITAL's standard file systems. P/OS supports RMS-11 files.

DCL requires that file specifications have the following format:

```
device:[directory]filename.type;version
```

The following example illustrates the format of a file specification:

```
BIGVOLUME:[USERFILES]MYFILE.TXT;15
  |         |         |   |   |
  device   directory name type version
```

The punctuation marks and brackets are required in order to separate the parts (fields) of the file specification. When you enter a file specification, you can omit fields and let DCL supply values for these fields. These DCL-supplied values are called defaults.

The fields and defaults for the file specification are:

device	The name of the disk or diskette that the file is on. This name must be a legal device or volume name. Some of the legal names for Professional storage devices include:
--------	--

BIGDISK:	(device name for hard disk)
BIGVOLUME:	(volume name for hard disk)
DISKETTE1:	(device name for diskette in slot 1)
DISKETTE2:	(device name for diskette in slot 2)

To get a complete list of all devices, use the **SHOW DEVICE** command. Your initial default device when you first install P/OS is **BIGVOLUME:** (hard disk systems) or **USERDISK:** (diskette systems). There are other device names with the format **ddnn:** (for example, **LP:** is the printer device

name). You should not need to use these unless directed to by specific application documentation.

You can specify a particular volume name instead of a device name. For example, to refer to a diskette you have named MEMOS, enter MEMOS:. Refer to Appendix C for more information on devices.

directory

The name of the directory the file is in. A directory name has one to nine characters (A–Z,0–9) and must also include the brackets in DCL directory specifications. You can use either braces ({ }), angle brackets (< >), or square brackets ([]). Your initial default directory when you first install P/OS is [USERFILES]. You can create new directories with the CREATE/DIRECTORY command. You can change your default directory with the SET DEFAULT command.

filename

The name of the particular file you are specifying. File names have one to nine characters (A–Z,0–9).

type

The file type of the particular file you are specifying. File types have three or fewer characters. Some file types are standard for certain applications. For example, the default file type for PROSE files is .DOC. Refer to application documentation for more information.

version

The version number of the file. The default is the most recent version.

Wildcards in File Specifications

You can use wildcards with the commands in this chapter, in addition to the regular defaults for the current device, the current directory, and the most recent version, to set up temporary defaults for every part of the file specification except the device name and the directory.

The simplest use of wildcards is the asterisk (*) to replace all or any field in the file specification.

For example, the * in place of the version number means “all versions.” The command

```
$ DELETE MYFILE.DOC;* <DO>
```

deletes all versions of the file MYFILE.DOC on the default volume and in the default directory.

The * can also be used to replace an entire file name or file type.

The command

```
$ DIRECTORY *.DOC <DO>
```

lists the most recent versions of all files with the type .DOC and any name on the default volume and in the default directory.

The command

```
$ DIRECTORY CHECKS.* <DO>
```

lists the most recent versions of all files with the name CHECKS and any type on the default volume and in the default directory.

The examples thus far have demonstrated the simple form of wildcarding, using the * to replace an entire field in a file specification.

With the APPEND, COPY, DELETE, DIRECTORY, PURGE, RENAME, TYPE, and UNLOCK commands, the * can be used in a more complex manner. The * actually means “match zero or all characters in this position.” For example, the command

```
$ DIRECTORY L*.TXT <DO>
```

lists the most recent versions of all files with the type .TXT whose names start with L on the default volume and in the default directory.

The command

```
$ DIRECTORY *L*.TXT <DO>
```

lists the most recent versions of all files with the type .TXT whose names include an L on the default volume and in the default directory.

The same substitutions can also be used in file types. For example, the command

```
$ DIRECTORY EXPENS.L* <DO>
```

lists the most recent versions of all files with the name `EXPENS` and the type beginning with an `L` on the default volume and in the default directory.

You can use more than one wildcard in file names and file types. For example, the command

```
$ DIRECTORY *F*D*.TXT <DO>
```

lists the most recent versions of all files with the type `.TXT` whose names include an `F` and a `D` in that order.

In addition, these same commands permit the percent sign (`%`) to be used as a wildcard, but only within file names and file types. The `%` means “match exactly one character in this position.”

For example, the command

```
$ DIRECTORY %.TXT <DO>
```

lists all files in the default directory with the type `.TXT` and a single-character file name.

The command

```
$ DIRECTORY NOV%%81.TXT <DO>
```

lists all files with the type `.TXT` and a file name consisting of `NOV` and `81` separated by two characters on the default volume in the default directory.

The wildcards can be combined in a single file specification. For example, the command

```
$ DIRECTORY %L*T.TM% <DO>
```

lists all files in the default directory with names beginning with a single character followed by an `L` and ending with a `T` and with a file type consisting of `.TM` and another single character.

Wildcarding, combined with systematic policies of directory assignments, file names, and file types can add considerable flexibility and convenience to your use of the system.

FILE PROTECTION

Each user has a unique User Identification Code (UIC), that is assigned when the account is set up. Your UIC identifies your account to the system.

UICs look like this:

[303,5]

In text and command descriptions, the UIC is often indicated by either of the following:

[g,m]

or

[UIC]

Where “g” indicates the group number and “m” is called the member number. Group and member numbers are octal and range from 1 through 377.

The UIC itself indicates whether or not you are privileged. Privileged users have group numbers from 1 through 10. Nonprivileged users have group numbers from 11 through 377.

When you create a file, you usually own it. Your UIC is an attribute of the file, identifying you as the owner. Each file also has a protection code. A protection code controls who can access a file and in what ways. You, as the file’s owner, control this protection code.

The file protection code specifies four categories of users, as well as four kinds of file access that each category of user can have. When you attempt to access a file, the operating system checks your UIC to determine which of the four user categories you belong to. Your ability to access the file is limited to the types of access that the file’s protection code grants to those categories.

The four categories of user in a file protection code are as follows:

SYSTEM	The operating system itself and privileged users. A privileged user has a UIC with a group number of 10 or less.
OWNER	The file owner is the user having the same UIC as the one the file was created under.
GROUP	All users having a UIC with the same group number as the one the file was created under.
WORLD	All other users.

The four kinds of file access that you can grant to the user categories are as follows:

- | | |
|--------|---|
| READ | The user, or the user's tasks, may read, copy, print, or type the file. |
| WRITE | The user, or the user's tasks, may add new data to the file by writing to it. You cannot edit a file without WRITE access, nor can you use the COPY command to write a file to a directory where you have no WRITE access. If the file is a task image file, WRITE access means you may run it. If you deny WRITE access to a class of users, they cannot run the task. |
| EXTEND | The user, or the user's tasks, may change the amount of disk space allocated to the file. In practice, EXTEND access is always specified along with WRITE access. |
| DELETE | The user, or the user's tasks, may delete the file. You cannot delete a file without DELETE access. |

Ordinarily, the operating system assigns the user's default protection code to files. In a protection code, the four types of file access are designated by their first letters. If the user does not change the default protection, the default will be:

SYSTEM:RWED,OWNER:RWED,GROUP:R,WORLD:

If you have files that you particularly want to protect against deletion, or if you want to limit the access people have to your files, use the SET PROTECTION or SET PROTECTION/DEFAULT commands, described in the sections on SET PROTECTION and SET PROTECTION/DEFAULT. Be aware, though, that there is no way to protect your files from privileged users, because they can change the protection code for any file.

You can display the protection for any file with the SHOW PROTECTION command. The DIRECTORY/FULL command also displays file protection, in the following form:

[RWED,RWED,RWED,R]

SPECIFYING FILES FOR NETWORK OPERATIONS

If you have PRO/DECnet installed on your Professional with the DCL extensions installed as described in the *PRO/DECnet User's Guide*, you can use DCL commands to operate on files at other nodes on your network. Node specifications must precede file specifications for files on remote nodes. A file descriptor with a node specification takes the following format:

```
[node specification::][file specification]
```

Node Specification

A node specification consists of the name of the node and optional access control information for that node, followed by two colons:

```
nodename[access-control]::
```

where:

nodename	is a 1- to 6-alphanumeric-character name that includes at least 1 alphabetic character. If a node name is an alias that includes associated access control information (a logical node name), you can omit all access control fields, as they will default to the information associated with the alias. For more information on aliases, see the <i>PRO/DECnet User's Guide</i> .
access-control	is information consisting of three position-dependent fields appended to a node name. Access control information can be specified in two ways: Format 1: /userid/passwd/accent:: Format 2: "userid passwd accent":: No embedded spaces within access control parameters are allowed in format 2. <i>userid</i> —is a 1- to 16-character string identifying the user at the remote system. For most systems, the <i>userid</i> is the same as the login id.

passwd—is a 1- to 8-character password needed to gain access to the remote file system. For most systems, the password is the same as that for logging in.

acct—is a 1- to 16-character string used to specify a billing account number at the remote system. (This field is mainly used by TOPS-10 and TOPS-20 systems.)

Both the interpretation of the access control fields and the access control mechanism depend on the type of remote system used.

Examples

Examples of valid node specifications follow. Missing fields may be omitted from the right. For example, the node specification QUEBEC/[310,2]// can be written QUEBEC/[310,2]. Missing fields on the left must be marked as missing by a slash or a blank (depending upon the format used), as shown in the fourth example, below.

```
NODE4/[7,7]/SECRET/ACCNT::
YUKON"5,10 LEFT"::
BOSS/EVERY/ONE::
BOSTON///ACCNTNE::
NODE1"RMES"::
BILBO/1,1/PRIV::
SHELOB/200,200/GUEST::
```

Foreign Files

Files that reside on nodes running operating systems other than RSX or P/OS are referred to as foreign files and must be syntax compatible with the systems on which they are located. Some foreign file formats are listed below. When using these or any other foreign format, enclose the foreign file directory and specification in double quotes (" "). This directs DCL to transmit the file specification to the foreign node without checking its syntax or applying defaults for missing fields. File specifications within double quotes are not used in determining default values for output files.

The use of wildcards in foreign file specifications is subject to the restrictions of the foreign operating system.

The next two tables list the directory and file specification formats for foreign files.

Table 2-1
Directory Format Table

<i>Operating System</i>	<i>Directory Format</i>
IAS	[ufd]
VMS	[directory]
RSTS/E	[ppn] or (ppn)
TOPS-20, VMS	<directory>
RT-11	(not applicable)

Table 2-2
File Specification Format Table

<i>Operating System</i>	<i>File Specification Format</i>
IAS, VMS	filename.type;ver
TOPS-20, VMS	filename.type.ver
RT-11, RSTS/E	filename.type

The following are examples of foreign file formats.

Table 2-3
Examples of Foreign File Formats

<i>Operating System</i>	<i>Example</i>
VMS	SYS__\$SYSDISK:[MITTON.DAP]DAPV70.MEM;69
TOPS-20	PS:<DECNETDEV>MAIL.TXT.1439
RSTS/E	SY:(1,4)SYSTEM.HLP
RT-11	DK:TECO.SAV

INDIRECT COMMAND FILES

You can execute a group of frequently used DCL commands more efficiently by creating an indirect command file and then *invoking* that file. Follow these steps:

1. Using PROSE or any other editor, create a file that consists of several DCL commands. For example,

```
$ PURGE [*]*.DOC
$ PURGE [*]*.BAS
$ DELETE *.TMP;*
$ APPEND/TODAY *.MEM MEMOS:[1985]MEMOS.FEB
$ TYPE VERIFY.TXT
$ SHOW DAYTIME
```

2. Give the file a name that describes generally what the commands will do, and give the file the type .CMD. (.CMD is the default file type for indirect command files.)

In the example above, the commands will purge all files of type .DOC and .BAS in directories on the default device; delete all files of type .TMP in the default directory; append all files of type .MEM edited today to the file MEMOS.FEB on a diskette volume named MEMOS: and directory named [1985]; display a file VERIFY.TXT containing some words such as “Your files have been cleaned up;” and, finally, display the date and time. For this example, the file could be called CLEANUP.CMD.

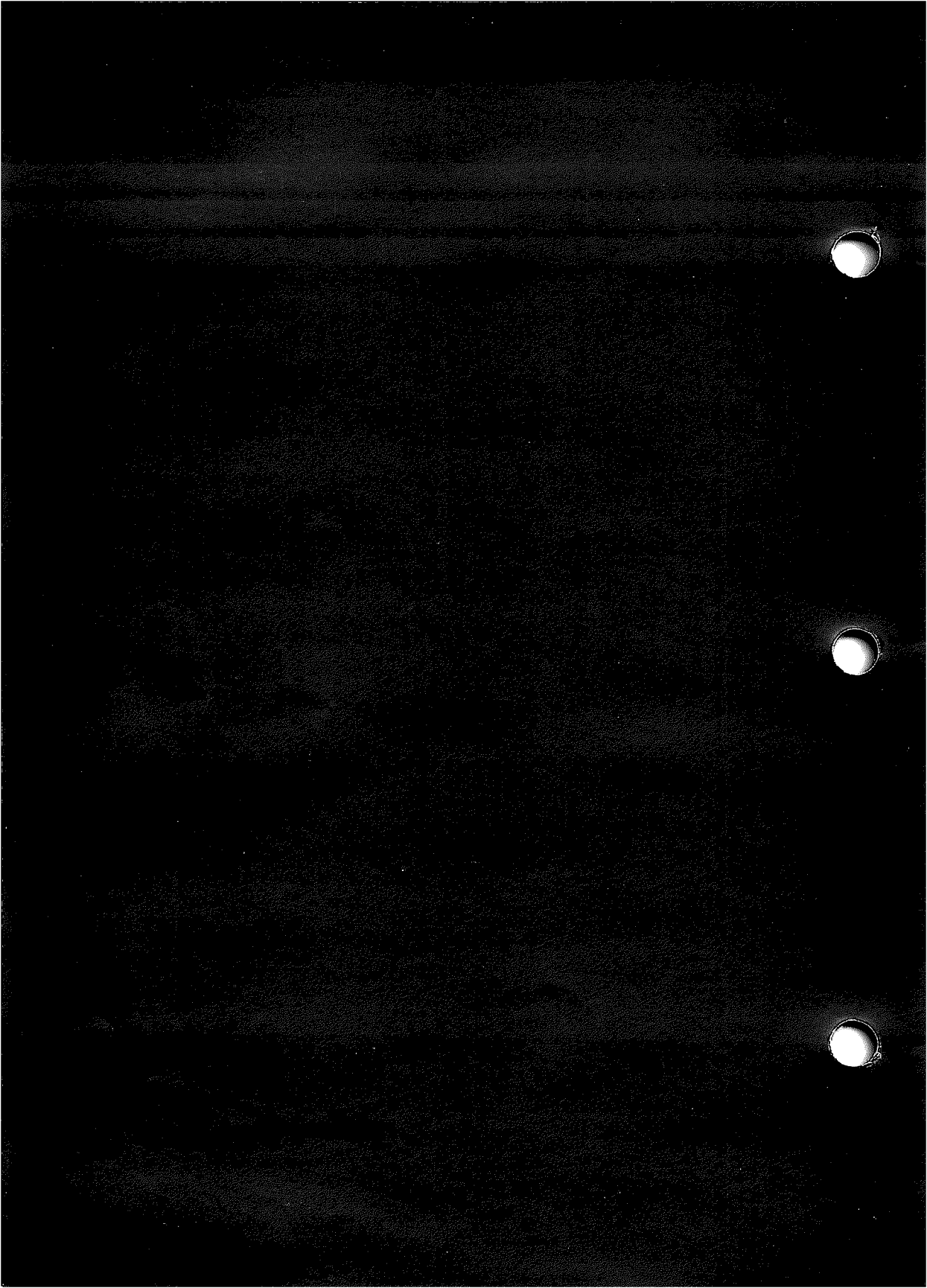
3. When you want to execute a command file, use the DCL format @FILENAME. For example:

```
$ @ CLEANUP <DO>
```

Command files can be very useful for lists of commands that you use frequently. If you set up rules for naming your files and directories, you can efficiently clean out files you do not need and prepare your own archives of old files you want to save. Applications you can buy for the Professional may also contain special command files that you may be instructed to invoke.

3

DCL Commands



Chapter 3

DCL Commands

This chapter lists the available DCL commands alphabetically, along with a description, format, example, qualifiers, and command prompts for each command.

The commands are:

ABORT	DIBOL	PRINT
APPEND	DIRECTORY	PRINT/REMOTE
ASSIGN	DISMOUNT	PURGE
ASSIGN/TASK	EDIT or EDIT/PROSE	REMOVE
BASIC	EDIT/EDT	RENAME
CANCEL	EXIT	RUN installed task
CLEAR	FIX	RUN uninstalled task
COBOL	FORMAT	SET
CONTINUE	FORTRAN	SHOW
COPY	HELP	SPAWN
CREATE	INITIALIZE	START
CREATE/DIRECTORY	INSTALL	START/UNBLOCK
DEASSIGN	MAIL	STOP/BLOCK
DEFINE	MOUNT	SUBMIT/REMOTE
DELETE	PASCAL	TYPE
DELETE/DIRECTORY	PHONE	UNLOCK

ABORT

ABORT forces an immediate end to a task that is running.

Format

ABORT taskname

taskname

The name of the task to be aborted.

Prompt

Task Name?

APPEND

APPEND attaches one or more sequential files, such as text files, to the end of an existing sequential file.

If you want to learn more about sequential files, consult *PRO/RMS-11: An Introduction* (Order No. AA-P098A-TK).

Format

```
APPEND[/qualifier[,...]] input-filespec[,...] output-filespec
```

Example

```
$ APPEND/SINCE:15-DEC-83 *.*;* MEMO01.DOC <DO>
```

qualifier

May be one or more of the following:

```
/DATE:dd-mmm-yy
/EXCLUDE:filespec
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/NOWARNINGS
```

With PRO/DECnet DCL extensions, the following qualifiers are also available:

```
/DATATYPE
/LOG
/MACY11
/PRINT
/TRANSFER__MODE
```

input-filespec

Specifies the file or files to be appended to the output file.

output-filespec

Specifies the file to which the input files are appended.

The output file must be an existing sequential file. The output file has the same version number after the APPEND command is executed, as it had before the command was issued. The input files appear at the end of the output file in the order they were specified.

No wildcards are permitted in the output file specification, and no qualifiers exist for it.

Although the output file must exist, it may be an empty file.

Prompts

File(s)?

To:

Qualifiers

/DATE:dd-mmm-yy

The **/DATE** qualifier specifies that you wish the APPEND command to affect only files created on the designated date.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the APPEND command not to affect certain files. The filespec argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/SINCE:dd-mmm-yy

The **/SINCE** qualifier specifies that you wish the APPEND command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the APPEND command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the APPEND command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **APPEND** command to affect only files created on the same day as the command is issued.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the “No such file” error messages should not be displayed when the input files specified do not exist.

Additional Qualifiers Available with PRO/DECnet**/DATA__TYPE**

The **/DATA__TYPE** qualifier specifies how the data in the file is interpreted. **/DATA__TYPE** with **APPEND** requires either the **ASCII** or **IMAGE** argument. The default is **IMAGE**. **/DATA__TYPE** must be entered after the file specification in the command line.

The **ASCII** argument transfers files in **ASCII** record mode and translates the records into an appropriate format for the remote system. You only need to use this argument when transferring text files to remote systems with file systems other than **FILES-11** or **RMS**. File transfers to **VAX/VMS** and **PDP-11/RSX** systems are more efficiently performed using the default, **IMAGE**.

The **IMAGE** argument transfers files with their current format and attributes. If a remote system cannot interpret the file format, an error message will be displayed. If the file is an **ASCII** text file, you can recover by retrying with the **ASCII** argument.

/LOG

The **/LOG** qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

/MACY11

The **/MACY11** qualifier specifies that the input or output file on the remote **TOPS-10** or **TOPS-20** system is in **MACY11** format. This informs the remote **DEC** system to handle the file appropriately when copying to or from the **P/OS** system. **/MACY11** must be entered after the file specification in the command line.

/PRINT

The **/PRINT** qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

/TRANSFER__MODE

The **/TRANSFER__MODE** qualifier specifies the method of packaging file data during a file transfer. This qualifier requires either the **BLOCK** or **RECORD** argument with the **APPEND** command. **RECORD** is the default. **/TRANSFER__MODE** must be entered after the file specification in the command line.

The **BLOCK** argument transfers files in block mode, using blocks of 512-byte sizes. This is more efficient than using record mode, but should only be used with systems that support **FILES-11** or **RMS-11**, such as **VAX/VMS**, **PDP-11/RSX**, or **PDP-11/RSTS/E RMS FAL**.

The **RECORD** argument transfers a file one record at a time. A record is a logical unit of data in a file.

ASSIGN

This command creates a logical name and assigns an equivalence string to the specified logical name. An equivalence name can be a device name, another logical name, a file specification, or any other string.

To specify the logical name table where you want to enter a logical name, use /USER, /SESSION, or /SYSTEM. If you do not explicitly specify a logical name table, the default is /SESSION.

Format

```
ASSIGN equiv__name logical__name [/qualifiers]
```

Examples

```
$ ASSIGN XX1:[CHARLES] CHARLIE: <DO>
$ PRINT CHARLIE:TEST.DAT <DO>
```

The ASSIGN command associates the logical name CHARLIE with the directory name [CHARLES] on the device XX1. Subsequent references to the logical name CHARLIE: result in the correspondence between the logical name CHARLIE: and the disk and directory specified. The PRINT command queues a copy of the file XX1:[CHARLES]TEST.DAT to the default print queue.

```
$ ASSIGN XX1:[CHARLES] CHARLIE: /USER <DO>
```

The ASSIGN command associates the logical name CHARLIE with the directory name [CHARLES] on the device XX1 and makes the assignment in the user logical name table.

```
$ ASSIGN/TRANSLATION__ATTRIBUTES=CONCEALED __DW002:[CHARLES.]-
$-PD001: <DO>
$ DIRECTORY PD001:[USERFILES] <DO>

Directory PD1:[USERFILES]
28-JUN-85 11:20

TEST.DAT 4. 27-JUN-85 17:21

Total of 4./4. blocks in 1. files
```

The ASSIGN command associates the logical name PD001: with the directory name [CHARLES] on the device DW002. Subsequent references to the device PD1: result in the device and directory specified as being used as the MFD for all file operations.

```
$ ASSIGN/TRANSLATION__ATTRIBUTES=TERMINAL XX001: TEMP: <DO>
```

The ASSIGN command associates the logical name TEMP: with the device XX001. Subsequent references to the logical TEMP: will result in the device XX001: being used without any further translation attempts on the equivalence name XX001:.

equiv__name

Defines the equivalence name to be associated with the logical name in the specified logical name table. The equivalence name string can contain from 1 to 255 characters. If the equivalence name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks (“ ”) in the places where you want a quotation mark to appear.

Note that if you enclose an equivalence name string in quotation marks, the case of alphabetic characters is preserved.

logical__name

Specifies the logical name string. The logical name string can contain from 1 to 255 characters. If the logical name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks (“ ”) in the places where you want a quotation mark to appear.

Note that if you enclose a logical name string in quotation marks, the case of alphabetic characters is preserved.

qualifiers

May be one or more of the following:

```
/USER
```

```
/SESSION
```

```
/SYSTEM
```

```
/FINAL
```

```
/TRANSLATION__ATTRIBUTES=[(keyword[,...])]
```

Prompts

```
Equivalence Name?
```

```
Logical Name?
```

Qualifiers**/USER**

Places the logical name in the user logical name table. Logicals created in the user logical name table are deleted when you exit any application (including DCL).

/SESSION

Places the logical name in the session logical name table. Session logical names are deleted when you log out of the system. If you do not explicitly specify a logical name table, the default is /SESSION.

/SYSTEM

Places the logical name in the system logical name table. All system users can access the logical name.

System logical names are never deleted unless an explicit deassign logical operation is performed.

/FINAL

Indicates that the equivalence name string should not be translated repeatedly; logical name translation should terminate with the current equivalence string. The /FINAL qualifier is synonymous with /TRANSLATION_ATTRIBUTES=TERMINAL.

/TRANSLATION_ATTRIBUTES=[(keyword[,...])]

Specifies one or more attributes that modify a logical name string. The /TRANSLATION_ATTRIBUTES qualifier is positional and must be specified before the equivalence name is specified.

You can specify the following keywords for translation attributes:

CONCEALED Indicates that the logical name is a concealed device name. If you specified the CONCEALED attribute, then the logical name must be a device name of the form DDNNN: and the equivalence name must be of the form _DDNNN:[directory.]. The device name in the equivalence name must be present. The [directory.] portion of the equivalence name is optional.

TERMINAL Indicates that the equivalence name string should not be translated iteratively; logical name translation should terminate with the current equivalence string.

If you specify only one keyword, you may omit the parentheses. Only the attributes you specify are set.



ASSIGN/TASK

ASSIGN/TASK reassigns the logical unit numbers (LUNs) of an installed task from one physical device to another. This reassignment overrides the static LUN assignments in the task's disk image file. ASSIGN/TASK cannot change the LUNs of an active task.

Format

ASSIGN/TASK:taskname ddn: lun

Example

```
$ ASSIGN/TASK: MYTASK TT2: 5 <DO>
```

taskname

Identifies the installed task whose LUN you wish to reassign.

ddn:

Specifies the new device to which you wish the LUN reassigned.

This can be a physical device, pseudo device, or logical device name.

lun

Specifies which LUN you wish to reassign to the new device.

Prompts

Task?

Device?

Logical Unit?

BASIC

BASIC invokes the PRO/Tool Kit BASIC-PLUS-2 compiler to begin a BASIC session. (Refer to the *BASIC on RSX-11M/M-PLUS Systems User's Guide* manual and the *BASIC Reference Manual* for additional information.)

Format

BASIC

Prompts

None

CANCEL

CANCEL eliminates entries from the clock queue. Either the RUN\$ directive or the time-based forms of the RUN command place entries in the clock queue. CANCEL only affects pending entries in the clock queue and does not affect a task that is currently executing.

Format

CANCEL taskname

taskname

The name of the inactive, installed task entry that is to be deleted from the clock queue.

Prompt

Task?

CLEAR

This command clears the screen, and optionally, performs a reset-to-initial state.

Format

`CLEAR [/qualifier]`

Prompts

None

Qualifier

`/RESET`

Performs a reset-to-initial state.

COBOL

COBOL invokes the PRO/Tool Kit COBOL-81 compiler to compile COBOL language source files.

NOTE: Please refer to the language documentation for additional information.

Format

COBOL[qualifier[s]] filespec

qualifiers

May be one or more of the following:

```

/[NO]ANSI__FORMAT
/[NO]CHECK[:arg]
        [NO]BOUNDS
        [NO]PERFORM
        ALL
        NONE
/COE:[NO]CIS
/[NO]CROSS__REFERENCE
/[NO]DEBUG
/[NOspec]
/[NO]LIST[:filespec]
/NAMES:xx
/[NO]OBJECT[:filespec]
/[NO]OVERLAY__DESCRIPTION
/[NO]SHOW:[NO]MAP
/[NO]SKELETON
/[NO]SUBPROGRAM
/TEMPORARY:device
/[NO]TRUNCATE
/[NO]WARNINGS:[NO]INFORMATIONAL

```

filespec

Specifies the source file to be input to the COBOL-81 compiler.

File input to the COBOL-81 compiler must contain COBOL source code. File specifications must have file names. The default file type is .CBL.

Prompt

File?

Qualifiers**/[NO]ANSI_FORMAT**

The **/ANSI_FORMAT** qualifier specifies that the source file is in conventional (or ANSI) format. Conventional format has 80-character lines with Area A beginning in character position 8.

The **/NOANSI_FORMAT** qualifier specifies that the source file is in terminal format, a DIGITAL-specified format that permits variable length lines with Area A beginning in character position 1.

The default is **/NOANSI_FORMAT**.

/[NO]CHECK[:arg]

[NO]BOUNDS
[NO]PERFORM
ALL
NONE

The **/CHECK** and **/CHECK:ALL** qualifiers are equivalent—both to each other, and also to **/CHECK:BOUNDS** and **/CHECK:PERFORM** in combination. They add object code that checks the ranges of subscripts, indexes, and nested **PERFORM** statements at run time.

The **/CHECK:BOUNDS** qualifier compares subscript and index ranges at run time against the ranges defined by corresponding **OCCURS** clauses. If any range is exceeded during program execution, COBOL-81 issues an error message.

The **/CHECK:PERFORM** qualifier determines whether or not your program's **PERFORM** statements are nested properly (if nested at all). If COBOL-81 detects improper nesting during program execution, it issues an error message.

The **/NOCHECK** and **/CHECK:NONE** qualifiers are equivalent—both to each other, and also to **/CHECK:NOBOUNDS** and **/CHECK:NOPERFORM** in combination. They suppress all range checking.

The **/CHECK:NOBOUNDS** qualifier suppresses range checking only for subscripts and indexes.

The `/CHECK:NOPERFORM` qualifier suppresses range checking only for nested `PERFORM` statements.

The qualifiers that suppress range checking reduce task size and improve program performance.

The default is `/CHECK (/CHECK:ALL)`.

`/CODE:[NO]CIS`

The `/CODE:CIS` qualifier tells the compiler to use CIS (Commercial Instruction Set) in the object code it produces.

NOTE: The Professional 300 series computers do not have the Commercial Instruction Set.

The `/CODE:NOCIS` qualifier tells the compiler not to use CIS.

These qualifiers override the default for your system. They are used to develop programs that will execute on a different system than the one used for program development. For example, if your system has CIS, you use `/CODE:NOCIS` to compile a program that will run on a system without CIS.

`/[NO]CROSS_REFERENCE`

The `/CROSS_REFERENCE` qualifier causes the compiler to produce a list file and to add two cross-reference tables to the end of the list file: one for data-names, and one for procedure-names. In each table, the names you use in your COBOL program are listed alphabetically. Opposite each name is a list of every line number in which that name occurs. A "D" after a number indicates the line in which you define the name. An asterisk (*) after a line number indicates a destructive reference, such as a value assignment to a data-name.

The `/NOCROSS_REFERENCE` qualifier suppresses production of the cross-reference tables.

The default is `/NOCROSS_REFERENCE`.

`/[NO]DEBUG`

The `/DEBUG` qualifier indicates that you plan to include the COBOL-81 Symbolic Debugger in your task image. To support the Symbolic Debugger, the compiler generates symbol information in the object module for all data-names and procedure-names in your program.

If you include the Symbolic Debugger in your program, you must also use the /DEBUG qualifier to the LINK command.

The /NODEBUG qualifier suppresses generation of symbol information in the object module.

The default is /NODEBUG.

[/NO]DIAGNOSTICS[:filespec]

The /DIAGNOSTICS qualifier creates a diagnostics file that contains the compiler diagnostic summary. If you do not append a file specification, the diagnostics file has the same file name as your source file, and the file type .DIA.

The /NODIAGNOSTICS qualifier suppresses the creation of a diagnostics file.

The default is /NODIAGNOSTICS.

[/NO]LIST[:filespec]

The /LIST qualifier produces a list file which contains both the complete source code and any diagnostic messages. If you do not append a file specification, the list file has the same file name as the source file and has the file type .LST.

The /NOLIST qualifier suppresses the production of a list file.

The default is /NOLIST.

/NAMES:xx

The /NAMES:xx qualifier tells the compiler to use the two alphanumeric characters you specify as the PSECT kernel for your program. You use this qualifier to ensure unique identification for PSECT kernels when your task image uses both subprograms and segmentation.

[/NO]OBJECT[:filespec]

The /OBJECT qualifier produces an object module. If you do not append a file specification to the qualifier, the object module has the same file name as the source, and the file type .OBJ.

The /NOOBJECT qualifier suppresses production of an object module and its associated skeleton ODL file. (See the description of [/NO]SKELETON.)

The default is /OBJECT.

`/[NO]OVERLAY_DESCRIPTION`

The `/OVERLAY_DESCRIPTION` qualifier produces two files—an indirect command file and an Overlay Descriptor Language file.

The indirect command file has the same file name as the source and the file type `.CMD`. It can be input to the Task Builder to create the task image.

The Overlay Descriptor Language file has the same file name as the source and the file type `.ODL`. It contains pointers to certain support routines that must be included in your task image (for example, system support for file input/output).

If you plan to use the `LINK/C81` command to task-build your program, do not use the `/OVERLAY_DESCRIPTION` qualifier. `LINK/C81` also creates a `.CMD` and an `.ODL` file, and cannot reference the `.CMD` and `.ODL` files created during program compilation.

The `/NOOVERLAY_DESCRIPTION` qualifier suppresses production of `.CMD` and `.ODL` files during program compilation.

The default is `/NOOVERLAY_DESCRIPTION`.

`/[NO]SHOW:[/NO]MAP`

The `/SHOW` and `/SHOW:MAP` qualifiers are equivalent. They cause the compiler to produce a list file and to append two offset maps to the list file. One offset map refers to the Data Division and one to the Procedure Division. The compiler provides these maps for use with the On-Line Debugging Tool (ODT). Consult the *IAS/R SX-11 ODT Reference Manual* for more information.

The `/NOSHOW` and `/SHOW:NOMAP` qualifiers are equivalent. They suppress production of the offset maps.

The default is `/NOSHOW (/SHOW:NOMAP)`.

`/[NO]SKELETON`

The `/SKELETON` qualifier produces a skeleton ODL file, which specifies the overlay structure for the object module. This file has the same file name as the source and the file type `.SKL`.

You cannot use the `/SKELETON` qualifier to override the `/NOOBJECT` qualifier. That is, an `.SKL` file cannot be produced when the `.OBJ` file is suppressed.

The `/NOSKELETON` qualifier suppresses production of the `.SKL` file.

You cannot use the `/NOSKELETON` qualifier if you plan to task-build your program with the `LINK/C81` command. `LINK/C81` assumes the presence of the `.SKL` file. Although the `.SKL` file is not strictly necessary for all programs, `DIGITAL` does not recommend that you suppress its production unless you are familiar with task-building alternatives to `LINK/C81`.

The default is `/SKELETON`.

`/[NO]SUBPROGRAM`

The `/SUBPROGRAM` qualifier tells the compiler to treat the source file as a subprogram. You should use this qualifier only if the subprogram does not use parameters from the main program; that is, if it does not contain the Procedure Division `USING` header.

The `/NOSUBPROGRAM` qualifier tells the compiler to treat the source as a main program.

The default is `/NOSUBPROGRAM`.

`/TEMPORARY:device`

The `/TEMPORARY` qualifier tells the compiler to store the temporary work files it uses during program compilation on the device you specify. Since the default device is the system disk (`SY:`), this qualifier is useful if there is little system disk space available.

`/[NO]TRUNCATE`

The `/TRUNCATE` qualifier tells the compiler to perform decimal truncation on the values of `COMP` data items. With decimal truncation, the maximum value the data item can contain depends on the item's `PICTURE` character-string.

The `/NOTRUNCATE` qualifier tells the compiler to perform binary truncation on the values of `COMP` data items. With binary truncation, the maximum value a `COMP` item can contain depends on its storage allocation.

The default is `/NOTRUNCATE`.

`/[NO]WARNINGS:[NO]INFORMATIONAL`

The `/WARNINGS` and `/WARNINGS:INFORMATIONAL` qualifiers are equivalent. They cause the compiler to issue informational, warning, and fatal diagnostics.

The `/NOWARNINGS` and `/WARNINGS:NOINFORMATIONAL` qualifiers are equivalent. They suppress production of informational, warning, and fatal diagnostics.

The default is `/WARNINGS (/WARNINGS:INFORMATIONAL)`.

CONTINUE

CONTINUE resumes the execution of a task that has been previously suspended.

Format

CONTINUE taskname

taskname

The name of the task to be restarted.

Prompt

Task Name?

COPY

COPY duplicates files.

You can also use COPY to copy two or more input files and create a single output file. The input files will be appended in the order entered. The output file will be a sequential file.

Format

```
COPY[/qualifier[,...]] input-filespec[,...] output-filespec
```

Examples

```
$ COPY PROG.BAS [GAMES] <DO>
```

In this example, PROG.BAS is copied to a directory named [GAMES].

```
$ COPY UTILACCT.DAT,TAXES.DAT YRLYEXP.DAT <DO>
```

In this example, DCL makes a copy of files UTILACCT.DAT and TAXES.DAT, appends TAXES.DAT to UTILACCT.DAT, and names the resulting file YRLYEXP.DAT.

qualifiers

May be one or more of the following:

```
/[NO]CONTIGUOUS
/REPLACE
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
/ALLOCATION:N[.]
/NONEW__VERSION
/OVERLAY
/PRESERVE__DATE
/SHARED
/OWN
```

With PRO/DECnet DCL extensions, the following qualifiers are also available:

```

/CONTIGUOUS
/CONCATENATE
/DATA__TYPE
/LOG
/MACY11
/NEW__VERSION
/NOSPAN
/PRINT
/SUBMIT
/TRANSFER__MODE

```

input-filespec

Specifies the input file or files to be copied. Multiple file specifications, separated by commas, are accepted. If you specify multiple input files, they will be concatenated in a single output file in the order that you specified.

output-filespec

Specifies a single output file to which the input file or files are copied. You can change the name, type, and version number of the file when you enter this parameter. Wildcards in the place of the name and the type leave the name and type unchanged. If you choose to use a wildcard in either of these fields, you must use a wildcard in both.

COPY always creates the output file. For example, if you type:

```
$ COPY FILE1.LIS FILE2.LIS DO
```

and FILE2 already exists, COPY will create a new version of the file, one higher than the existing version. If FILE2 does not already exist, COPY will create a file with the name FILE2 and type .LIS.

If you specify a version number for the output file field, then a file of that version number is created. If such a file already exists, the operation fails.

Wildcards are acceptable for output files if the destination is another directory. If you have multiple input files and use wildcards for the output file, you create multiple output files, each with the name and type of the corresponding input file.

You can send copies to devices, for example, the printer (LP:), as well as to directories.

You can also use the COPY command to create multiple copies of the same file with identical or different names.

Prompts

From?
To?

Qualifiers

/[NO]CONTIGUOUS

Specifies that the output file must be contiguous. A contiguous file is one that is stored on the disk or diskette in such a way that each block of the file is in a continuous series. Certain kinds of files, such as task image – .TSK – files, can only be used properly if they are contiguous. This qualifier can be used, for example, to make a task file contiguous. To see if a file is contiguous, use the DIRECTORY command. The letter “C” appears before the file creation date if the file is contiguous.

If this qualifier is not used, then only files that are already contiguous remain contiguous when copied.

/REPLACE

If the output file has the same name, type, and version number as an already existing file at the destination, the first file is deleted and the file you have sent replaces it. The name, type, and version number stay as they were.

/DATE:dd-mmm-yy

The /DATE qualifier specifies that you wish the COPY command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The /SINCE qualifier specifies that you wish the COPY command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **COPY** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **COPY** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **COPY** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **COPY** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

/ALLOCATION:N[.]

Specifies that **N** blocks of contiguous space be allocated for the new copy of the file. Unless the decimal point is appended, **N** is octal.

/NONEW_VERSION

Prevents the version number from being automatically incremented when a file is copied. You must specify the **/NONEW_VERSION** qualifier when you copy all versions of a particular file name and type.

/OVERLAY

Causes the contents of the input file or files to be copied into the output file. Whatever was in the output file before is destroyed, but the file ID is unchanged. The output file must exist before you issue the command.

/PRESERVE_DATE

Specifies that you want the output file to have the same creation date as the input file. The default is to have today's date as the creation date of the input file.

/SHARED

Specifies that you want other users to be able to access the file while you are copying it.

/OWN

Changes the ownership of the file being copied to the destination directory. After execution, both directories own their respective copies. If you do not specify **/OWN**, the original UIC owns both copies. If you are copying from another directory to your own, use this qualifier.

Additional Qualifiers Available with PRO/DECnet**/CONTIGUOUS**

The **/CONTIGUOUS** qualifier specifies that the output file must be contiguous. If this qualifier is not used, then only files that are already contiguous remain contiguous when copied. **/CONTIGUOUS** must appear after the file specification in the command line.

/CONCATENATE

The **/CONCATENATE** qualifier specifies that the output file should be a single file that contains all of the input files in the order specified. If wildcards are used in the input file specification, the files will be concatenated in random order.

/DATA_TYPE

The **/DATA_TYPE** qualifier specifies how the data in the file is interpreted. With the **COPY** command, the **DATA_TYPE** qualifier takes either the **ASCII** or **IMAGE** argument. The default is **IMAGE**. **/DATA_TYPE** must appear after the file specification in the command line.

The ASCII argument transfers files in ASCII record mode and translates the records into an appropriate format for the remote system. You only need to use this argument when transferring text files to remote systems with file systems other than RMS. File transfers to VAX/VMS and PDP-11/RXS systems, which use RMS, are more efficiently performed using the default, IMAGE.

The IMAGE argument transfers files with their current format and attributes. If a remote system cannot interpret that format, an error message will be displayed. If the file is an ASCII text file, you can recover by retrying with the ASCII argument.

/LOG

The /LOG qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

/MACY11

The /MACY11 qualifier specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACY11 format. This informs the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACY11 must appear after the file specification in the command line.

/NEW_VERSION

The /NEW_VERSION qualifier specifies that the output files will be created with a version number higher than any existing files of the same name in the directory.

/NEW_VERSION must appear after the file specification in the command line.

/NOSPAN

The /NOSPAN qualifier specifies that the output file will be created with records that do not span block boundaries. Normally, only files that already have records that do not span block boundaries remain as such when copied. /NOSPAN must appear after the file specification in the command line.

/PRINT

The **/PRINT** qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

/SUBMIT

The **/SUBMIT** qualifier specifies that the output file is a temporary file and will be submitted to the remote batch processor facility when copied. The file will be deleted after it is processed at the remote node.

/TRANSFER__MODE

The **/TRANSFER__MODE** qualifier specifies the method of packaging file data during a file transfer. For the **COPY** command, the **TRANSFER MODE** qualifier takes the argument **AUTOMATIC**, **BLOCK**, or **RECORD**. **AUTOMATIC** is the default. **/TRANSFER__MODE** must appear after the file specification in the command line.

The **AUTOMATIC** argument causes DCL to select either **BLOCK** or **RECORD** after checking the file capabilities at the remote node. If you have problems transferring files with the **AUTOMATIC** argument, try specifying either **BLOCK** or **RECORD**.

The **BLOCK** argument transfers files in block mode, using blocks of 512-byte sizes. This is more efficient than using record mode, but should only be used with systems that support RMS-11, as VAX/VMS, PDP-11/RXS, or PDP-11/RSTS/E RMS FAL.

The **RECORD** argument transfers a file one record at a time. A record is a logical unit of data in a file.

CREATE

CREATE creates a sequential file and enables you to type text directly into the file from your terminal without using an editor.

Format

CREATE filespec

filespec

Specifies the name of the file to be created.

As soon as the command is entered, the cursor moves down a line. The file is open for input. Any text you type goes into the file.

When you have finished entering text, press **CTRL/Z** to close the file.

If you wish to create an empty sequential file, simply enter the **CTRL/Z** first.

If you use **CTRL/U** when creating a sequential file, the text on the line is eliminated, but not the line itself. In other words, **CTRL/U** leaves a blank line behind when it deletes a line. **CTRL/R** repaints the current line you are typing. **CTRL/U**, **CTRL/R**, and the **DELETE** key are the only editing facilities available to you when creating sequential files at the terminal.

Prompt

File?

Additional Qualifiers Available with PRO/DECnet

With PRO/DECnet DCL extensions, the following qualifiers are available:

```

/CONTIGUOUS
/LOG
/MACY11
/NEW_VERSION
/NOSPAN
/NOWARNINGS
/PRINT
/REPLACE
/SUBMIT

```

With PRO/DECnet, the CREATE command creates a file from records you type. In other words, when you press DO after entering the CREATE command, all subsequent keystrokes are entered into a file that you named with the CREATE command. To close and store the file, press CTRL/Z.

Command Qualifiers

/LOG

The /LOG qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

/NOWARNINGS

The /NOWARNINGS qualifier specifies that the “No such file” error messages should not be displayed when the input files specified do not exist.

/PRINT

The /PRINT qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

/SUBMIT

The /SUBMIT qualifier specifies that the output file is a temporary file and will be submitted to the remote batch processor facility when copied. The file will be deleted after it is processed at the remote node.

File Qualifiers

/CONTIGUOUS

The CONTIGUOUS qualifier specifies that the output file must be contiguous. /CONTIGUOUS must appear after the file specification in the command line.

/MACY11

The /MACY11 qualifier specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACY11 format. This informs the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACY11 must appear after the file specification in the command line.

`/NEW_VERSION`

The `/NEW_VERSION` qualifier specifies that the output files will be created with a version number higher than any existing files of the same name in the directory.

`/NEW_VERSION` must appear after the file specification in the command line.

`/NOSPAN`

The `/NOSPAN` qualifier specifies that the output file will be created with records that do not span block boundaries. Normally, only files that already have records that do not span block boundaries remain as such when copied. `/NOSPAN` must appear after the file specification in the command line.

`/REPLACE`

With the `/REPLACE` qualifier, if the output file has the same name, type, and version number as an already existing file at the destination, the first file is deleted and the file you have sent replaces it. The name, type, and version number stay as they were. `/REPLACE` must appear after the file specification in the command line.

CREATE/DIRECTORY

CREATE/DIRECTORY creates a User File Directory (UFD) on a FILES-11 volume and enters the UFD into the volume's Master File Directory (MFD).

Format

CREATE/DIRECTORY [ddnn:][directory][qualifier(s)]

[ddnn:][directory]

You must specify at least the directory parameter. If you do not specify a device or volume, the default volume is used.

ddnn:

Specifies the name of the device on which you want the directory created. The default is SY0.

[directory]

Specifies the name of the directory in which you wish to create.

Example

```
$ CREATE/DIRECTORY SYSDISK:[DIRECTORY] <DO>
```

qualifiers

May be one or more of the following:

```
/NOWARNINGS
/OWNER__UIC:[UIC]
```

Prompt

Directory?

Qualifiers

/NOWARNINGS

Suppresses error messages resulting from the command.

`/OWNER_UIC:[UIC]`

Specifies the User Identification Code [UIC] of the directory's owner. A UIC establishes the protection status of the directory. See the section on "File Protection" for more information.

The default is for you to own the directory. You can specify any UIC as the owner. However, a nonprivileged user cannot always create files in a directory that is owned by a UIC different from his own.

DEASSIGN

DEASSIGN deletes a logical name assignment previously made with the ASSIGN command (see ASSIGN).

If you do not specify any qualifier, the search for the logical is limited to the SESSION logical name table.

CAUTION: Logical names may also be created through P/OS and under program control. DO NOT deassign any of these. Use this command only to deassign logical names you have created with the ASSIGN command.

Format

DEASSIGN logicalname [/qualifier(s)]

Example

```
$ DEASSIGN JOE <DO>
```

logical__name

The logical to be deassigned. The logical name can also be a quoted string.

qualifiers

May be one or more of the following:

```
/USER
/SESSION
/SYSTEM
/ALL
```

Prompt

Logical Name?

Qualifiers

/USER

Limits the search for the logical name to the user logical name table. The user logical name table contains temporary logicals that are deleted by the system when you exit the DCL application.

/SESSION

Limits the search for the logical name to the session logical name table. The session logical name table contains logical names that will remain in the system until you log out.

/SYSTEM

Limits the search for the logical name to the system logical name table. The system logical name table contains logicals that everyone logged into the system has access to. System logicals are removed either by using the DEASSIGN command, or when the system is powered down.

/ALL

Deassigns all of the logicals in the table specified.

DEFINE

This command creates a logical name and assigns an equivalence string to the specified logical name. An equivalence name can be a device name, another logical name, or a file specification.

To specify the logical name table where you want to enter a logical name, use /USER, /SESSION, or /SYSTEM. If you do not explicitly specify a logical name table, the default is /SESSION.

Format

```
DEFINE logical__name equiv__name [/qualifiers]
```

Examples

```
$ DEFINE CHARLIE: XX1:[CHARLES] <DO>
$ PRINT CHARLIE:TEST.DAT <DO>
```

The DEFINE command associates the logical name CHARLIE with the directory name [CHARLES] on the device XX1. Subsequent references to the logical name CHARLIE: result in the correspondence between the logical name CHARLIE: and the disk and directory specified. The PRINT command queues a copy of the file XX1:[CHARLES]TEST.DAT to the default print queue.

```
$ DEFINE CHARLIE: XX1:[CHARLES] /USER <DO>
```

The DEFINE command associates the logical name CHARLIE with the directory name [CHARLES] on the device XX1 and makes the assignment in the user logical name table.

```
$ DEFINE/TRANSLATION__ATTRIBUTES=CONCEALED PD001:
__DW002:[CHARLES.] <DO>
```

```
$ DIRECTORY PD001:[USERFILES] <DO>
```

```
Directory PD1:[USERFILES]
28-JUN-85 11:20
TEST.DAT 4, 27-JUN-85 17:21
Total of 4./4. blocks in 1. files
```

The DEFINE command associates the logical name PD001: with the directory name [CHARLES] on the device DW002. Subsequent references to the device PD1: result in the device and directory specified as being used as the MFD for all file operations.

```
$ DEFINE/TRANSLATION__ATTRIBUTES = TERMINAL TEMP: XX001: <DO>
```

The **DEFINE** command associates the logical name **TEMP:** with the device **XX001**. Subsequent references to the logical **TEMP:** will result in the device **XX001:** being used without any further translation attempts on the equivalence name **XX001:**.

qualifiers

May be one or more of the following:

```
/USER
```

```
/SESSION
```

```
/SYSTEM /FINAL
```

```
/TRANSLATION__ATTRIBUTES = [(keyword[,...])]
```

logical__name

Specifies the logical name string. The logical name string can contain from 1 to 255 characters. If the logical name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks (" ") in the places where you want a quotation mark to appear.

Note that if you enclose a logical name string in quotation marks, the case of alphabetic characters is preserved.

equiv__name

Defines the equivalence name to be associated with the logical name in the specified logical name table. The equivalence name string can contain from 1 to 255 characters. If the equivalence name contains quotation marks, enclose the string in quotation marks and use two sets of quotation marks (" ") in the places where you want a quotation mark to appear.

Note that if you enclose an equivalence name string in quotation marks, the case of alphabetic characters is preserved.

Prompts

```
Logical Name?
```

```
Equivalence Name?
```

Qualifiers**/USER**

Places the logical name in the user logical name table. Logicals created in the user logical name table are deleted when you exit any application (including DCL).

/SESSION

Places the logical name in the session logical name table. Session logical names are deleted when you log out of the system. If you do not explicitly specify a logical name table, the default is /SESSION.

/SYSTEM

Places the logical name in the system logical name table. All system users can access the logical name.

System logical names are never deleted unless an explicit deassign logical operation is performed.

/FINAL

Indicates that the equivalence name string should not be translated iteratively; logical name translation should terminate with the current equivalence string. The /FINAL qualifier is synonymous with /TRANSLATION_ATTRIBUTES=TERMINAL.

/TRANSLATION_ATTRIBUTES=[(keyword[,...])]

Specifies one or more attributes that modify a logical name string. The /TRANSLATION_ATTRIBUTES qualifier is positional and must be specified before the logical name is specified.

You can specify the following keywords for translation attributes:

CONCEALED Indicates that the logical name is a concealed device name. If you specified the CONCEALED attribute, then the logical name must be a device name of the form DDNNN: and the equivalence name must be of the form _DDNNN:[directory.]. The device name in the equivalence name must be present. The [directory.] portion of the equivalence name is optional.

TERMINAL Indicates that the equivalence name string should not be translated iteratively; logical name translation should terminate with the current equivalence string.

If you specify only one keyword, you may omit the parentheses. Only the attributes you specify are set.

DELETE

DELETE removes a file specification entry from a directory, and makes available the disk or diskette space taken by the file(s) for reuse by other files.

Format

```
DELETE[/qualifier[,...]] filespec[,...]
```

Example

```
$ DELETE/TODAY *.TMP;* <DO>
```

qualifier

May be one or more of the following:

```
/LOG
/QUERY
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
```

filespec

Specifies the file or files to be deleted. You must specify the name, file type, and version number fields of the file specifications you wish to delete. Device and directory fields default to your current device and directory. DELETE accepts wildcards in any field except the device field.

You need not supply a file type to delete a file with a null file type.

Prompt

```
File(s)?
```

Qualifiers

/LOG

Specifies that a list of the files deleted be displayed on your screen.

/QUERY

Specifies that you wish to decide which files should be deleted on an individual basis. Each file specified in the command is named. You may enter one of four characters:

Y—Deletes file named and goes on to next file.

N—Does not delete file named and goes on to next file.

G—Deletes the file and goes on to delete all other files specified.

Q—Does not delete the file and exits the command. No more files are deleted.

Remember that you can specify files by default or wildcard. If you do not specify a version number, then /QUERY is the default.

/DATE:dd-mmm-yy

The /DATE qualifier specifies that you wish the DELETE command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The /SINCE qualifier specifies that you wish the DELETE command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The /THROUGH qualifier specifies that you wish the DELETE command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The /SINCE and /THROUGH qualifiers can be combined to specify that you wish the DELETE command to affect only files created within the range of designated dates.

/TODAY

The /TODAY qualifier specifies that you wish the DELETE command to affect only files created on the same day as the command is issued.

`/EXCLUDE:filespec`

The `/EXCLUDE` qualifier specifies that you wish the `DELETE` command not to affect certain files. The `filespec` argument to `/EXCLUDE` can contain wildcards, but the file specification must contain a version number, either explicitly or as the `*` wildcard.

`/NOWARNINGS`

The `/NOWARNINGS` qualifier specifies that the “No such file” error message should not be displayed when an input file specified does not exist.

DELETE/DIRECTORY

This command deletes an empty directory. An error message appears if you attempt to delete a directory that is not empty.

NOTE: Be sure that the directory that you are deleting is not your default directory.

Format

DELETE/DIRECTORY [device]:[directory-name]

NOTE: The brackets around the device indicate that the device is optional for this command. The brackets around the directory-name are part of the directory-name syntax.

directory-name

The name of the empty directory you want to delete.

Example

```
$ DELETE/DIRECTORY [USERFILES] <DO>
```

Prompt

Directory?

DIBOL

This command invokes the PRO/Tool Kit DIBOL compiler to compile DIBOL source code files.

NOTE: Please refer to the language documentation for additional information.

Format

DIBOL[/qualifier[s]] filespec[,s][/qualifier[,s]]

qualifier

May be one or more of the following:

```

/[NO]BUILD
/[NO]CROSS__REFERENCE
/[NO]DEBUG
/[NO]LIST[ = filespec]
/[NO]OBJECT[ = filespec]
/[NO]OPTIMIZE
/PAGE__SIZE = number (D:system default)
/[NO]STANDARD
/[NO]SHOW
/[NO]WARNINGS

```

filespec

is the name of a DIBOL source code file.

To reference a universal library containing source code modules, use the following format:

DIBOL[/qualifiers[s]] filespec[,s][/qualifier[,s]] librspec/LIBR

librspec

is the name of the universal (default) library.

/LIBR

signals the DIBOL compiler that librspec is a universal (default) library. You may include only one such library in each compilation.

Prompt

File(s)?

Qualifiers

/[NO]BUILD

/BUILD causes the build option to be invoked automatically after the compilation phase. This qualifier creates .CMD and .ODL files for direct input to the Professional Application Builder (PAB).

/[NO]DEBUG

/DEBUG includes extra information for the DIBOL debugger. The default is /NODEBUG which prevents the inclusion of this information.

/[NO]LIST[= filespec]

/LIST causes the DIBOL to output a file with the same name as the source listing. /LIST creates this output. You may change the name of this output file by typing:

/LIST = new-filespec

Refer to NOTE in /[NO]OBJECT.

/[NO]OBJECT

/OBJECT creates an object module with the same name as the source listing and is the default value. /OBJECT = new-filespec creates an object file with a file name different than the source file name. /NOOBJECT suspends the creation of an object file. (When you list more than one source file on the command line and you specify /OBJECT, DIBOL creates an object file with the name of the first source file listed.)

NOTE: Use of the qualifiers /NOOBJECT and /NOLIST allows the DIBOL command to check for errors.

/OPTIMIZE

/OPTIMIZE causes DIBOL to optimize the object file. The default is /NOOPTIMIZE which does not produce an optimized object file.

/[NO]STANDARD

/STANDARD flags all occurrences of P/OS (RSX-11M-PLUS) extensions to DIBOL-83. This is the default.

/SHOW = (arg[,...])

The following table gives the arguments for the **/SHOW** qualifier.

NOTE: In the following table, D indicates default value for the qualifier.

Table 3-1
Arguments for the /SHOW Qualifier

<i>Argument</i>	<i>Default</i>	<i>Description</i>
ALL		Shows everything.
NONE		Shows only source (nothing else).
CON	D	Shows non-compiled conditionals.
NOCOND		Does not show non-compiled conditionals.
INCLude	D	Shows included files.
NOINCL		Does not show included files.
TABLEs		Shows symbol and label tables.
NOTABL	D	Does not show symbol and label tables.
SOURce	D	Shows DIBOL source.
NOSOUR		Does not show DIBOL source.

/[NO]WARNINGS

/WARNINGS displays (or prints if accompanied by **/LIST**) warning messages produced by the DIBOL compiler; **/WARNINGS** is the default value. **/NOWARNINGS** suppresses the display (listing) of warning messages.

DIRECTORY

DIRECTORY displays information for an individual file, or a group of files.

Some of the information displayed will be meaningful only to experienced DCL users. This book does not explain every item in each type of directory display.

Format

```
DIRECTORY[/format-qualifier][/destination-qualifier]
          [/other-qualifier[,...]] [filespec[,...]]
```

Example

```
$ DIRECTORY/FULL PROGA.BAS <DO>
```

format-qualifier

Controls the appearance of the directory list.

```
/FULL
/BRIEF
/FREE [device:]
/SUMMARY
```

PRO/DECnet allows the following additional qualifiers:

```
/ATTRIBUTES
/WIDTH:n
```

destination-qualifier

Controls the destination of the command.

```
/OUTPUT:filespec
```

other-qualifier

Modifies the list of files by creation date or exclusion, and may be one or more of the following:

```
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
```

filespec[,...]

Specifies the file or files for which information should be displayed. If you do not supply a file specification, a complete listing for the current directory is displayed.

If you supply one or more file specifications, separated by commas, then directory information on those files is displayed.

You can use wildcards in place of any file specification field except the device field. If you do not supply a version number, only information on the most recent versions is displayed. However, if you do not supply a file type, a null file type is assumed. If you do not know the file type, use a wildcard.

You can display a listing for another directory by supplying the directory name in this field. You can also specify one of the device or volume names described in Chapter 2, File Specifications.

If you name files in two directories, you should name files for the current directory first. If you name files from another device or directory first, the defaults are canceled.

Prompts

None

Qualifiers

There are three groups of qualifiers:

Format qualifiers—control the appearance of the directory.

Destination qualifiers—control where to send the command output.

Other qualifiers—modify the list of files by creation date or exclusion.

If you do not supply a qualifier from the format-qualifier list, the display is in standard format, giving the file name, type, version number, the number of blocks the file occupies, and the date and time of creation. Also, a C indicates a file is contiguous and an L indicates a file is locked.

If you do not include a qualifier from the destination-qualifier list, the display appears on your screen.

If you do not include any qualifier from the other-qualifier list, the display includes all files that otherwise qualify.

Format Qualifiers

/BRIEF

The **/BRIEF** qualifier specifies that the directory listing include file names, types, and version numbers only.

/FULL

The **/FULL** qualifier specifies that the complete directory listing include the standard display information as well as the owner UIC, file ID number, blocks used and allocated, and protection status of the file.

/SUMMARY

The **/SUMMARY** qualifier specifies that the display give only the total number of blocks allocated and used for the specified files. If you give no file specifications in the command, the display shows the total blocks allocated and used for the default directory.

/FREE [device:]

The **/FREE** qualifier specifies that the display give the free space, largest contiguous free space, and number of free file headers on the default device or a specified device.

Format Qualifiers Additionally Available with PRO/DECnet

/ATTRIBUTES

The **/ATTRIBUTES** qualifier displays the information included in a **FULL** directory listing, plus the file organization, record format, and record attributes for each file.

/WIDTH:n

The **/WIDTH** qualifier specifies the width of the display for the default and **/FULL** format displays only. The default value is 72. At least one field is always displayed per line, even if that field exceeds the maximum width specified.

Destination Qualifiers

/OUTPUT:filespec

Specifies that the output of the **DIRECTORY** command be placed in a file having the file specification you supply.

Other Qualifiers

/DATE:dd-mmm-yy

The **/DATE** qualifier specifies that you wish the **DIRECTORY** command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The **/SINCE** qualifier specifies that you wish the **DIRECTORY** command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **DIRECTORY** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **DIRECTORY** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **DIRECTORY** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **DIRECTORY** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the **"*"** wildcard.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the "No such file" error message should not be displayed when an input file specified does not exist.

DISMOUNT

This command marks the volume mounted on the specified device to be logically off-line and disconnected from the file system.

Format

```
DISMOUNT devname [volumelabel]
```

Example

```
$ DISMOUNT DISKETTE1: <DO>
```

devname

The name of the device on which the magnetic medium containing the volume is currently mounted. This argument can be in the form `ddnnn:`, or it can be a logical name previously assigned to the device by the `MOUNT` command.

volumelabel

The name of the volume. If the volume currently mounted in `devname` does not match `volumelabel`, the system does not dismount the device.

Prompt

```
Device name?
```

EDIT OR EDIT/PROSE

EDIT/PROSE invokes the PROSE editor from DCL.

NOTES: This command only applies to the hard disk system. On diskette systems, use the PROSE application diskette.

For hard disk systems, this section only describes the DCL command that accesses the PROSE editor. For further information on PROSE, refer to your User's Guide.

If you prefer to use PRO/EDT, refer to the EDIT/EDT command in this manual and to the *PRO/EDT User's Guide* for the Professional.

This section only describes the DCL command that accesses the PROSE editor. For further information on PROSE you should turn to the *Hard Disk System User's Guide*.

Format

EDIT/PROSE/qualifier input-filespec

Example

```
$ EDIT CHECKS.BAS <DO>
```

qualifier

/OUTPUT:output-filespec

input-filespec

The name of an existing input file that PROSE will edit or a new filespec that PROSE will first create and then edit.

Prompt

File?

Qualifier

`/OUTPUT:output-filespec`

Causes EDIT/PROSE to name the output file with a different file name than the input file name.

NOTE: PROSE accessed as a separate application (from the Main Menu or Additional Applications Menu) gives you a choice of saving or not saving the file following the EXIT command. However, when you EXIT from a DCL editing session, the file is automatically saved with any changes from that session. EDIT creates a file specification of the same name and type, but with a version number one higher than the highest existing version number for the input file. If you do not want to save the output file, you have to delete it after exiting the editing session.

EDIT/EDT

EDIT/EDT starts an interactive editing session with the PRO/EDT editor to create or modify a file.

Format

EDIT/EDT[/qualifier[s]] filespec

Example

\$ EDIT/EDT filename <DO>

qualifier

May be one or more of the following:

/[NO]COMMAND[:filespec]
/[NO]JOURNAL[:filespec]
/[NO]OUTPUT[:filespec]
/[NO]RECOVER
/[NO]READ_ONLY

filespec

The name of an existing file or the name of a file to be created. You must specify a file name, but the file type may be null.

Prompt

File?

NOTE: EDIT/EDT may return the following error:

Sorry, task not installed

If this occurs and you want to use the PRO/EDT editor, refer to Appendix B to install the PRO/EDT editor.

Qualifiers

/[NO]COMMAND[:filespec]

Controls whether an PRO/EDT initialization file is read by PRO/EDT before editing begins. These files contain commands that alter the default setup for PRO/EDT, such as custom line-mode commands and change-mode key definitions.

The default is /COMMAND:LB:[1,2]EDTSYS.EDT. If you use this qualifier and EDTINI.EDT or some other file you name does not exist, PRO/EDT issues no error message and continues with the editing session.

If you do not wish to use LB:[1,2]EDTSYS.EDT, use the /NOCOMMAND qualifier.

/[NO]JOURNAL[:filespec]

Controls whether PRO/EDT creates a journal file for the editing session. The default is to create a journal file with a file name the same as that of the input file with the type .JOU. You can specify a different name by including a filespec, EDIT/EDT/JOURNAL:TEMPORARY.JOU REPORT.MEM, for example.

The journal file consists of all editing commands and text entered during the session. If the editing session ends abnormally, such as through a system crash, or if you inadvertently type three CTRL/Zs in succession, the journal file is saved. In such a case, invoke PRO/EDT again, with the same command line as before plus the /RECOVER qualifier. Your editing session is repeated and all your editing is restored.

If the editing session ends normally, the journal file is deleted.

If you specify /NOJOURNAL, no journal file is created and no recovery is possible.

/[NO]OUTPUT[:filespec]

To alter the name of the output file include a filespec with the /OUTPUT qualifier.

If you do not specify this qualifier, the default is to create a file of the same name and type as the input file with a version number one higher than the input file. If the file is new, PRO/EDT creates version number 1.

If you specify /NOOUTPUT, you *must* include a file specification in your EDT EXIT command in order to exit EDT.

`/[NO]READ_ONLY`

Specifies whether you wish simply to read the file or to edit it. If your command line includes `/READ_ONLY`, you can use the full facilities of PRO/EDT, but you cannot exit without including a filespec in your PRO/EDT EXIT command. Normally, you would use the PRO/EDT QUIT command if you had specified `/READ_ONLY`. The `/READ_ONLY` qualifier is equivalent to a combination of `/NOOUTPUT` and `/NOJOURNAL`. You can use `/READ_ONLY` to look at files to which you have no write access.

The default is `/NOREAD_ONLY`, which need never be specified.

`/[NO]RECOVER`

Specifies whether PRO/EDT reads commands from a journal file prior to starting the editing session. With a journal file, your editing session can be restored if interrupted by a system crash or other problem. The default is `/NORECOVER`, which need never be specified. The `/RECOVER` qualifier requests PRO/EDT to open the input file and then read PRO/EDT commands and text from the file with the same file name as the input file and the file type `.JOU`. The command line with `/RECOVER` added to it must be identical to the command line that initiated the original failed editing session. This means that if you specified a PRO/EDT initialization file, you must specify the same file in the `/RECOVER` command line. And, if you specified a name for the journal file other than `infile.JOU`, you must include the `/JOURNAL` qualifier with the appropriate filespec. If journaling was not enabled on the original command line, you cannot recover the editing session.

EXIT

This command causes you to exit from DCL. Pressing the EXIT key also causes you to exit.

Format

EXIT

Prompts

None

FIX

This command loads and locks a task into its partition. Subsequent requests for running the task are serviced more quickly because the task is memory-resident and does not have to be loaded from the disk before it can run.

Fixed tasks remain physically in memory even after they exit. Therefore, they do not have to be reloaded when a request is made to run them. (Note that reexecuting fixed tasks is not always practical; to reexecute, the task must initialize all impure data and LUN assignments at run time.) Only a REMOVE command can free the occupied memory partition.

Note the following requirements:

- You must install a task before you fix it.
- You cannot fix a task whose name is in the form ...xxx or xxx\$\$\$.
- This is because the copy of the task with that form of name is a nonexecutable, prototype task.
- You cannot fix an active task.
- You should not fix a checkpointable task because doing so makes the task noncheckpointable.

See also the INSTALL command with the /FIX qualifier.

Format

FIX taskname [/qualifier]

taskname

The task or region to fix in memory.

qualifier

May be the following:

/REGION or

/READONLYSEGMENT

Prompt

Task?

Qualifier**/REGION**

Indicates that the task or common to be fixed is a region, such as a read-write common.

/READONLYSEGMENT

Indicates that the task or common is to be fixed as read-only.

FORMAT

This command sets up a hard disk so that it is recognized by its controller. Once you have formatted a disk, you can then use the INITIALIZE command to initialize it.

Normally, the P/OS installation procedure formats the system disk. However, to connect additional hard disks to a Professional you must format each one by invoking the FORMAT command.

Format

```
FORMAT devname [qualifier]
```

devname

The name of the device to be formatted. This argument can be in the form ddnn[:], or it can be a logical name previously assigned to the device in the MOUNT command.

qualifier

May be the following:

```
/NOCONFIRM
```

Prompt

```
Device Name?
```

Qualifier

```
/NOCONFIRM
```

This qualifier suppresses the confirmation of the format request if the volume is already in FILES-11 format.

FORTRAN

FORTRAN invokes the PRO/Tool Kit FORTRAN-77 compiler to compile FORTRAN language source files.

NOTE: Please refer to the language documentation for additional information.

Format

FORTRAN[/qualifier[s]] filespec[s]

qualifier

May be one or more of the following:

```

/[NO]CHECK
/CONTINUATIONS:n
/[NO]DEBUG
/[NO]DLINES
/IDENTIFICATION
/[NO]I4
/[NO]LIST[:filespec]
/[NO]MACHINE_CODE
/[NO]MAP
/[NO]OBJECT[:filespec]
/[NO]OPTIMIZE
/[NO]SHAREABLE
/SOURCE
/[NO]STANDARD:arg
    ALL
    NONE
    SOURCE
    SYNTAX
/[NO]TRACEBACK:arg
    ALL
    BLOCKS
    LINES
    NAMES
    NONE
/WORK_FILES:n

```

filespec

The name of the FORTRAN source file.

Prompt

File(s)?

Qualifiers

/[NO]CHECK

The /CHECK qualifier specifies that you wish the compiler to check that all array references are within bounds.

The default is /NOCHECK.

/CONTINUATIONS:n

The /CONTINUATIONS qualifier specifies the maximum number of continuation lines permitted in the code. The value of n can be from 0 through 99. The default is 5.

/[NO]DEBUG

Specifies that the compiler is to provide symbol table information for use by the PRO/Tool Kit FORTRAN-77 symbolic debugger. When you use the /DEBUG command qualifier, you should also use the /NO-OPTIMIZE command qualifier. /NODEBUG is the default.

/[NO]DLINES

The /DLINES qualifier specifies that lines with a "D" character in column 1 are to be compiled. The default is /NODLINES, meaning these lines are treated as comment lines. Use this feature for debugging.

/IDENTIFICATION

The /IDENTIFICATION qualifier causes the compiler to print its identification and version number on your terminal.

/[NO]I4

The /I4 qualifier causes a two-word default allocation for integer variables. The default is /NOI4, meaning the compiler causes a single-word allocation for any integer variables not given an explicit length specification.

[NO]LIST[:filespec]

Specifies whether a compiler listing should be generated. The default is `/NOLIST`, meaning no compiler listing is generated.

If you do not supply a file specification for this qualifier, the listing has a file name derived from the name of the first source file in the FORTRAN command with the file type `.LST`. If you wish the listing to have a different name, supply the name as an argument to the `/LIST` qualifier. If you do supply a name, the listing file appears in your directory but is not printed on the printer.

The `/LIST` qualifier behaves in a different manner depending on whether it is given as a command qualifier or a filespec qualifier. If `/LIST:LP:` is used as a command qualifier, the listing file is placed in your directory and printed on the printer. If `/LIST` is used as a filespec qualifier, the listing file appears in your directory but is not printed on the printer. If `/LIST` is used as a filespec qualifier, the listing file takes the name of that file.

NOTE: The `/MACHINE_CODE`, `/MAP`, and `/SOURCE` qualifiers all affect the contents of the compiler listing file.

[NO]MACHINE_CODE

The `/MACHINE_CODE` qualifier specifies that you wish the compiler listing to include binary machine code and diagnostics. The default is `/NOMACHINE_CODE`.

`/MACHINE_CODE` implies `/LIST`. You do not need to include the `/LIST` qualifier unless you wish to use it to establish a name for the listing file different from any of the file names included in the compilation.

The `/MACHINE_CODE` qualifier behaves in a different manner depending on whether it is given as a command qualifier or a filespec qualifier. If `/MACHINE_CODE` is used as a command qualifier, the listing file is placed in your directory and printed on the printer. If `/MACHINE_CODE` is used as a filespec qualifier, the listing file appears in your directory but is not printed on the printer. If `/MACHINE_CODE` is used as a filespec qualifier, the listing file takes the name of that file.

/[NO]MAP

The /MAP qualifier specifies that you wish the compiler listing to include a storage map and diagnostics.

/MAP implies /LIST. You do not need to include the /LIST qualifier unless you wish to use it to establish a name for the listing file different from any of the file names included in the compilation.

The /MAP qualifier behaves in a different manner depending on whether it is given as a command qualifier or a filespec qualifier. If /MAP is used as a command qualifier, the listing file is placed in your directory and printed on the printer. If /MAP is used as a filespec qualifier, the listing file appears in your directory but is not printed on the printer. If /MAP is used as a filespec qualifier, the listing file takes the name of that file.

/[NO]OBJECT[:filespec]

Determines whether or not an object module is generated by the compiler. The default is /OBJECT, which does create a module. The default name of the object file created by FORTRAN is the name of the last-named source file with the file type .OBJ. If you wish the object file to have a different name, supply the name as an argument to the /OBJECT qualifier.

/NOOBJECT specifies that no object module is created. You can use the /NOOBJECT qualifier to get a compiler listing file to check for errors without generating object code.

/[NO]OPTIMIZE

Controls whether the compiler optimizes the compiled program to generate more efficient code.

Use /NOOPTIMIZE in conjunction with the /DEBUG qualifier to link a P/OS FORTRAN-77 program with the debugger so that variables always contain their updated values.

/SHAREABLE

The /SHAREABLE qualifier states that you wish the compiler to generate pure code and pure data sections as read-only. This takes advantage of code sharing in multiuser tasks on P/OS.

/SOURCE

The /SOURCE qualifier specifies that you wish the compiler listing to include the source code.

/SOURCE implies /LIST. You do not need to include the /LIST qualifier unless you wish to use it to establish a name for the listing file different from any of the file names included in the compilation.

The /SOURCE qualifier behaves in a different manner depending on whether it is given as a command qualifier or a filespec qualifier. If /SOURCE is used as a command qualifier, the listing file is placed in your directory and printed on the printer. If /SOURCE is used as a filespec qualifier, the listing file appears in your directory but is not printed on the printer. If /SOURCE is used as a filespec qualifier, the listing file takes the name of that file.

/[NO]STANDARD:arg

ALL
NONE
SOURCE
SYNTAX

The /STANDARD qualifier directs the compiler to look in your source code for extensions to ANSI standard (X3.9-1978) FORTRAN at the full-language level. If the compiler finds extensions, it flags them and produces informational diagnostics.

The ALL argument produces informational diagnostics for all detected extensions.

The NONE argument produces no informational diagnostics.

The SOURCE argument produces informational diagnostics for lowercase letters and tab characters in the source code.

The SYNTAX argument is the same as /STANDARD with no argument.

The default is /NOSTANDARD.

`/[NO]TRACEBACK:arg`
ALL
BLOCKS
LINES
NAMES
NONE

`/TRACEBACK` controls the amount of extra code included in the compiled output for use by the OTS during error traceback. This code is used in producing diagnostic information and in identifying which statement in the FORTRAN source program caused an error condition to be detected during execution.

The ALL argument states that error traceback information is to be compiled for all source statements, and function and subroutine entries.

The LINES argument is the same as ALL.

The BLOCKS argument states that traceback information is to be compiled for subroutine and function entries and initial statements in sequences called blocks. This is the default.

The NAMES argument states that traceback information is to be compiled only for subroutine and function entries.

The NONE argument states that no traceback information is to be produced.

`/WORK__FILES:n`

The `/WORK__FILES` qualifier specifies the number of temporary on-disk files you wish used during the compilation. Where `n` can be from 0 through 3; the default is 2. Increasing the number of work files increases the maximum possible size of your program but decreases the speed of compilation.

HELP

HELP displays useful information about most of the commands, qualifiers, and other options used in DCL. Extensive HELP is only available on hard disk systems.

Format

```
HELP [topic] [/qualifier1] [/qualifier2]
```

Example

```
$ HELP /DIRECTORY /FULL <DO>
```

topic

May be one or more of the following:

APPEND	brief	COPY
CREATE	DELETE	DIRECTORY
EDIT	EXIT	filespec
mistakes	PURGE	RENAME
SET	SHOW	TYPE
UNLOCK		

qualifier

May be one or more of the following:

```
/OUTPUT:filespec
/FILE:filespec
/filename
/LOCAL
```

```
qualifier1, qualifier2
```

Optional qualifiers that provide additional information on the scope of the DCL command. Available qualifiers are listed at the bottom of each help screen.

Prompts

None

Qualifiers

/OUTPUT:filespec

Permits you to name an output file where the requested help text is to be saved. The default is /OUTPUT:TI:.

/FILE:filespec

Specifies any file where help text is located. If you do not give a complete file specification, the defaults are LB:[1,2]filename.HLP.

/filename

Specifies that the help text begins with LB:[1,2]filename.HLP.

/LOCAL

Specifies that the help text is in the file HELP.HLP in the default directory on the default device.

INITIALIZE

This command deletes all existing files on a volume, imparts the standard FILES-11 file structure. Note that the volume must be formatted (see the **FORMAT** command) prior to being initialized.

Format

INITIALIZE[/qualifier] devname [volumelabel]

devname

The name of the device on which the magnetic medium containing the volume is currently mounted.

This argument can be in the form `ddnn[:]`, or it can be a logical name previously assigned to the device in the **MOUNT** command.

volumelabel

The label that names the volume after initialization.

It can be from 1 to 12 characters long.

Prompts

Device name?

Volume name?

Qualifiers

/[NO]BAD__BLOCKS

Performs a check for bad blocks on the volume you are initializing. If any bad blocks are found, P/OS will not use them for storing information. The default is to perform a bad check on the volume.

/NOCONFIRM

Suppresses the message confirming the initialization request if the volume is already in FILES-11 format.

INSTALL

INSTALL includes a specific task in P/OS, thus making it known to the system. An installed task is dormant until you request it to run with the RUN command.

INSTALL is not the same as “Install application” on the P/OS Menu for installing applications. You should not use the INSTALL command unless directed to a specific application.

Format

```
INSTALL[/qualifier[,...]] filespec
```

Example

```
$ INSTALL APP.TSK <DO>
```

qualifier

May be one or more of the following:

```
/FIX
/NOREMOVE
/READONLY__COMMON
/TASK__NAME:taskname
/[NO]WRITEBACK
```

[\$]filespec

Specifies the name of the task image file containing the task you wish to install. .TSK is the default file type.

The dollar sign (\$), if present, directs the system to search for the file in the application directory containing DCL. If you do not include the /TASK__NAME qualifier, the task will be installed under a name based on the first six characters of the file name unless another name was assigned when the task was built.

Prompt

File?

Qualifiers**/FIX**

Loads and locks a task into memory after installing it.

/NOREMOVE

P/OS will not remove tasks or regions that have the /NOREMOVE qualifier.

/READONLY_COMMON

Specifies that a common region is to be installed as a read-only common.

/TASK_NAME:taskname

Specifies the name by which the task is to be referenced. The default is set at link time. This qualifier overrides the link-time specification.

/[NO]WRITEBACK

Applies only when you install a read-write common. This qualifier controls whether or not the system writes the common back to its task image either when you remove the common from memory or when the system checkpoints the common.

If you use /WRITEBACK, then any changes made to the common while installed are copied to the common's task image when the common is checkpointed or you remove it.

If you specify /NOWRITEBACK, then changes made to the common are not copied back to the image. The default (that is, if you do not specify either) is /NOWRITEBACK.

MAIL

Invokes the PRO/DECnet Mail Utility that allows you to send or read messages to or from other users on your system, or another system connected to your system by PRO/DECnet. For a complete description of the Mail Utility see the *PRO/DECnet User's Guide*.

Format

MAIL

Prompts

None

Qualifiers

None

MOUNT

This command declares a volume (and therefore the files or data it contains) to be logically known to the system, and available for use.

Format

MOUNT ddnn: [/qualifier]

ddnn

The device on which you want to mount the volume. You can optionally specify a colon at the end of the device name.

qualifier

May be any combination of the following:

/FOREIGN

/SHOW

Prompt

Device name?

Qualifiers

/FOREIGN

Specifies that the volume being mounted is not in standard P/OS (FILES-11) format.

/SHOW

Displays the volume label after mounting the volume.

PASCAL

This command invokes the PRO/Tool Kit PASCAL compiler to compile one or more PASCAL source programs.

NOTE: Please refer to the language documentation for additional information.

Format

PASCAL[/qualifier[s]] filespec[,s]

qualifier

May be one or more of the following:

```

/[NO]CHECK[:arg]
    ALL
    BOUNDS
    SUBRANGE
    POINTER
    STACK
    DIVIDE
    CASE

/CODE = arg
    FPP
    EIS

/[NO]DEBUG = TRACEBACK
/[NO]LIST[ = filespec]
/[NO]MACHINE_CODE[ = filespec]
/[NO]OBJECT[ = filespec]
/[NO]STANDARD
/[NO]XRAY
/[NO]LOG
  
```

filespec

The name of a PASCAL source file.

Prompt

Files?

Qualifiers**/[NO]CHECK[:arg]**

The **/CHECK** option controls run-time checking.

Arguments:

ALL	enables all values (default)
BOUNDS	enables array bounds checking
CASE	enables CASE statement expression checking
DIVIDE	enables divide by zero checking
POINTER	enables NIL pointer checking
STACK	enables stack overflow checking
SUBRANGE	enables subrange checking

/CODE = arg

The **/CODE** option controls whether the code generated by Tool Kit PASCAL uses FPP instructions to perform floating point operations or EIS instructions to simulate floating point operations with subroutine calls.

Arguments:

FPP	generates FPP instructions (default)
EIS	generates EIS instructions

/[NO]DEBUG = TRACEBACK

The **/DEBUG = TRACEBACK** option controls whether Tool Kit PASCAL generates code that tracks the relationship between the lines in the program source code and the current position in the executing program. When **/DEBUG = TRACEBACK** is enabled, run-time error messages indicate the line which caused the error and the currently active procedure and function invocations.

/[NO]LIST[= filespec]

The **/LIST[= filespec]** option controls whether Tool Kit PASCAL generates a program listing file. The default is **/NOLIST**; thus compiler error messages appear on your terminal. If you specify **/LIST** and do not supply a file specification, the default is **file.LST**, where **file** is the name of the first input file.

/[NO]MACHINE_CODE[= filespec]

The **/MACHINE_CODE** option controls whether Tool Kit PASCAL generates a MACRO-11 source file containing the assembly language code for the program. The default is **/NOMACHINE_CODE**. The default name of the macro file created by PASCAL is the name of the last-named source file with the file type **.MAC**. If you want the macro file to have a different name, supply the name as an argument to the **/MACHINE_CODE** qualifier.

You can examine or assemble this file.

Also, you must not specify the **/OBJECT** and **/MACHINE_CODE** qualifiers together on the same command line.

/[NO]OBJECT[= filespec]

Determines whether or not an object module is generated by the compiler. The default is **/OBJECT**, which does create a module. The default name of the object file created by PASCAL is the name of the last-named source file with the file type **.OBJ**. If you want the object file to have a different name, supply the name as an argument to the **/OBJECT** qualifier.

/NOOBJECT specifies that no object module is created. You can use the **/NOOBJECT** qualifier to get a compiler listing file to check for errors without generating object code.

You must not specify the **/OBJECT** and **/MACHINE_CODE** qualifiers together on the same command line.

/[NO]STANDARD

The **/STANDARD** option flags any uses of language features not supported by the ISO PASCAL standard. The default is **/NOSTANDARD**.

/[NO]XRAY /[NO]LOG

The **/XRAY** or **/LOG** option causes Tool Kit PASCAL to display its progress as it compiles a program. The default is **/NOXRAY** and **/NOLOG**.

PHONE

Invokes the PRO/DECnet Phone Utility that allows you to communicate with other users on your system, or another system connected to your system by PRO/DECnet. For a complete description of the Phone Utility, including information on the `PHONE` command line, see the *PRO/DECnet User's Guide*.

Format

`PHONE` [phone-command]

Prompts

None

PRINT

This command allows printing to the local printer. The current default settings will be used to print the file(s).

Format

PRINT filename

Example

```
$ PRINT BOOK.MEM <DO>
```

filename

The name of the file(s) to print.

Prompt

File(s)?

PRINT/REMOTE

PRINT/REMOTE is available only with PRO/DECnet. PRINT/REMOTE directs an existing file on a remote node to a line printer for printing. If the file you want to print is not on that node, you may wish to use the COPY command with a /PRINT qualifier.

Format

```
PRINT/REMOTE[/qualifier[,...]] printfile[,...]
```

Example

```
$ PRINT REMOTE LOG MYFILE.DOC <DO>
```

qualifier

May be one or more of the following:

```
/LOG
```

```
/NOWARNINGS
```

printfile

The file descriptor (node specification and file specification) for each file to be queued for printing.

Prompt

```
Files?
```

Qualifiers

/LOG

The /LOG qualifier specifies that the names of files printed are to be displayed as the operation is performed.

/NOWARNINGS

The /NOWARNINGS qualifier specifies that the “No such file” error messages should not be displayed when the input files do not exist.

PURGE

PURGE deletes all but the latest versions of files, and releases the storage space that the deleted files occupied.

Format

```
PURGE[/qualifier[,...]] filespec[,...]
```

Example

```
$ PURGE/EXCLUDE:MYFILE.TXT *.* <DO>
```

qualifier

May be one or more of the following:

```
/[NO]LOG
/KEEP:n
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
```

filespec

The name of the file, whose old versions are to be deleted. The file must not be protected from delete.

Prompt

```
File(s)?
```

Qualifiers

```
/[NO]LOG
```

The /NOLOG qualifier specifies that the files deleted by PURGE be listed on your screen. The default is /NOLOG.

/KEEP:n

The **/KEEP** qualifier specifies that the *n* latest versions of a file be retained.

If you do not use this qualifier, all versions except the most recent of a given file, are deleted. That is, the default form of the command includes the qualifier **/KEEP:1**. With the qualifier explicitly stated, all but the *n* highest numbered versions are deleted. **PURGE** assumes that the version numbers of files are in sequential order, for example:

```
FILE1.LIS;4 FILE1.LIS;3 FILE1.LIS;2 FILE1.LIS;1 ...
```

and that there are no versions missing from the sequence, for example:

```
FILE1.LIS;20 FILE1.LIS;15 FILE1.LIS;14 ...
```

If more than one file specification is given with the **/KEEP** qualifier, all but the latest *n* versions of all files listed are deleted.

/DATE:dd-mmm-yy

The **/DATE** qualifier specifies that you wish the **PURGE** command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The **/SINCE** qualifier specifies that you wish the **PURGE** command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **PURGE** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **PURGE** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **PURGE** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **PURGE** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the "No such file" error messages should not be displayed when the input files do not exist.

REMOVE

This command removes the name of a task from the System Task Directory. The task is then no longer installed. REMOVE/REGION takes the name of a region out of the Common Block Directory and the partition list. REMOVE is not the same as “Remove Application” on the P/OS Menu.

CAUTION: You should not use the REMOVE command unless directed to for a specific application.

Format

REMOVE[qualifier] taskname

Example

```
$ REMOVE APP.TSK <DO>
```

qualifier

/REGION

taskname

The name of the task to be removed.

Prompt

Task?

Qualifier

/REGION

The /REGION qualifier specifies that you wish to remove a region from the Common Block Directory.

RENAME

RENAME changes the name, type, or version number of an existing file.

Format

RENAME[/qualifier[...]] old-filespec new-filespec

Example

* \$ RENAME/THROUGH:15-FEB-85 BOOK.DOC (OLD)BOOK.DOC <DO>

qualifier

May be one or more of the following:

/DATE:dd-mmm-yy

/SINCE:dd-mmm-yy

/THROUGH:dd-mmm-yy

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

/TODAY

/EXCLUDE:filespec

/NOWARNINGS

PRO/DECnet allows the following additional qualifiers:

/LOG

/NEW_VERSION

old-filespec

The file name prior to renaming.

new-filespec

The desired new name of the file. The file does not get copied. Therefore, if you want to specify a new directory for the file, you can only specify a directory that is on the same volume.

If you have PRO/DECnet and are renaming a remote file, access control information should be given for the old-filespec only.

* REN OLDNAME NEWNAME <DO>

Prompts

Old file name?
New file name?

Qualifiers

/DATE:dd-mmm-yy

The **/DATE** qualifier specifies that you wish the **RENAME** command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The **/SINCE** qualifier specifies that you wish the **RENAME** command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **RENAME** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **RENAME** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **RENAME** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **RENAME** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/NOWARNINGS

Specifies that the "No such file" error messages should not be displayed when the old-filespec does not exist.

Additional Qualifiers Available with PRO/DECnet

/LOG

Specifies that the names of files renamed are to be displayed as the operation is performed.

/NEW__VERSION

Specifies that the output files will be created with a version number higher than any existing files of the same name in the directory. /NEW__VERSION must be entered after the file specification in the command line.

RUN installed task

If the task is installed, RUN initiates its execution. You can use RUN to initiate the execution of installed tasks on a schedule by creating entries in the system clock queue.

Format

```
RUN[/qualifier[,...]] taskname
```

Example

```
$ RUN APP <DO>
```

qualifier

May be one or more of the following:

```
/COMMAND:"taskcommand"
```

```
/DELAY:nu
```

```
/INTERVAL:nu
```

```
/SCHEDULE:hh:mm:ss
```

```
/SYNCHRONIZE:u
```

taskname

The name of the task that RUN will execute.

Prompt

```
Task?
```

Qualifiers

```
/COMMAND:"taskcommand"
```

Use the /COMMAND qualifier to pass a command to the task you are running. The command must be inside the quotes. For example, the command

```
$ RUN/COMMAND:"PIP /LI" $PIP <DO>
```

runs PIP (Peripheral Interchange Program) and then passes the /LI (LIST) switch (qualifier) to it. This results in a directory listing.

/DELAY:nu

Specifies that the task be run after the stated amount of time passes. The argument *nu* specifies the amount of the delay as a number of units. Where *n* is the number of units and *u* is the time unit as follows:

- T – Ticks
- S – Seconds
- M – Minutes
- H – Hours

Your system has a programmable clock. The frequency of this clock is 64 Hz, which results in a tick length of 1/64th of a second.

Acceptable values for these units are as follows:

- T – Any positive value to a maximum of 32767.
- S – Any positive value to a maximum of 32767.
- M – The maximum value is 1440.
- H – The maximum value is 24.

The system always waits at least one interval. If you specify 0, the system treats it as a 1.

/INTERVAL:nu

Specifies that the task is to be run at regular intervals. The argument *nu* specifies the interval as a number of units of time. Where *n* is the number of units, and *u* is the unit as follows:

- T – Ticks
- S – Seconds
- M – Minutes
- H – Hours

See the **/DELAY** qualifier for a detailed description of these units.

/SCHEDULE:hh:mm:ss

Specifies that the task be run at a particular time of day.

/SYNCHRONIZE:u

Specifies that the execution of the task be synchronized on the next occurrence of a particular clock unit. Where u is the time unit as follows:

- T - Ticks
- S - Seconds
- M - Minutes
- H - Hours

RUN uninstalled task

This command checks to see if the task is installed. If the task is not installed, RUN installs the task image, runs it, and removes the name of the task from the System Task Directory upon task completion.

Format

```
RUN[/qualifier] [$]filespec
```

Example

```
$ RUN APP.TSK <DO>
```

qualifier

May be one or more of the following:

```
/COMMAND:"taskcommand"
```

```
/TASK_NAME:taskname
```

[\$]filespec

The name of the uninstalled task image. The dollar sign (\$) directs the system to search first for the file in the DCL application directory.

Prompt

```
Task?
```

Qualifiers

```
/COMMAND:"taskcommand"
```

Use the /COMMAND qualifier to pass a command to the task you are running. The command must be inside the quotes. For example, the command:

```
$ RUN/COMMAND:"PIP /LI" $PIP <DO>
```

runs PIP (Peripheral Interchange Program) and then passes the /LI (LIST) switch (qualifier) to it. This results in a directory listing.

`/TASK__NAME:taskname`

Specifies the name under which the task is to be run. The default is the name of the terminal.

Task names are restricted to six Radix-50 characters. The Radix-50 character set consists of the 26 uppercase letters, the numbers 0 through 9, the dollar sign (\$), and the period (.).

SET [DAY]TIME

This command sets the current date and time.

Format

SET [DAY]TIME:[dd-mmm-yy] [hh:mm]

Example

```
$ SET TIME:8-SEP-83 14:25 <DO>
```

dd-mmm-yy

Specifies the date. You can enter the date in either of two formats:

dd-mmm-yy Where dd is the number of the day, mmm is the first three characters of the name of the month, and yy is the number of the year (relative to 1900).

mn/dd/yy Where mn is the number of the month, dd is the number of the day, and yy is the number of the year (relative to 1900).

Regardless of the format you choose, the date is displayed in the first format.

hh:mm[:ss]

The time to which the system will be set (24-hour format) in the form 14:35.

Prompts

Time?

Date?

SET DEFAULT

The SET DEFAULT command establishes the current directory.

Format

```
SET DEFAULT [device:][directory]
```

NOTE: The brackets around the device indicate that the device is optional for this command. The brackets around the directory-name are part of the directory-name syntax.

Example

```
$ SET DEFAULT DISKETTE1:[LETTERS] <DO>
```

device:

The device name.

directory

The directory name enclosed in square brackets.

Prompt

Device and/or directory?

SET DEVICE

SET DEVICE establishes certain attributes for the specified device.

Format

```
SET DEVICE:ddnn:/device-attribute[s]
```

ddnn

Specifies the device for which attributes are to be set. When the one-line form of the command is used, the parameter is ddnn: preceded by a colon (:) at the end of the word DEVICE.

device-attribute

May be one or more of the following:

```
/[NO]LOWERCASE  
/WIDTH:n
```

Example

```
$ SET DEVICE:TT2:/NOLOWERCASE <DO>
```

Prompts

```
Device name?  
Device attribute?
```

Device Attributes

```
/[NO]LOWERCASE
```

The /LOWERCASE attribute sets a terminal or line printer so that lowercase characters are not converted to uppercase for printing.

The /NOLOWERCASE attribute sets a terminal or line printer so that lowercase characters are converted to uppercase for printing. This is the default.

You can use the SET TERMINAL/LOWERCASE command to set TI: in this fashion.

/WIDTH:n

Sets the size of a device's I/O buffer. The value of *n* (octal) is the length in characters of a line on the device.

For terminals, *n* must be greater than 2 and not greater than 255. The terminal driver does not discard excess characters, but puts them in a record of their own. That is, excess characters appear one line below the line in which they should appear.

SET HOST

With PRO/DECnet only, SET HOST invokes the Remote DECnet Terminal application (available with PRO/DECnet). This allows your Professional node to communicate with a remote host, performing terminal operations as though it were local to the host. The host must support terminal emulation from remote nodes.

Format

```
SET HOST node-name
```

node-name

The name of the node to connect to.

Example

```
$ SET HOST EDEN <DO>
```

Prompts

None

Qualifiers

None

To disconnect from a VMS or RSX host computer, either log off of the host, or use the MAIN SCREEN or EXIT key. It is preferable to log off of the host computer.

To disconnect from a TOPS-10 or TOPS-20 host computer, log off of the host, and then press the MAIN SCREEN or the EXIT key.

Once you have exited from the host computer, press RESUME to return control to your node.

Refer to the *PRO/DECnet User's Guide* for information about differences in processing during terminal emulation.

SET PRIORITY

This command alters the priority of an active task.

Format

```
SET PRIORITY:n taskname
```

n

Specifies the new priority you wish to assign to the task. Priority numbers are in the range of from 1 through 250.

taskname

The name of the task whose priority is to change.

Example

```
$ SET PRIORITY:1 MYTASK <DO>
```

Prompts

```
Priority number?  
Task Name?
```

SET PROTECTION

Use SET PROTECTION to change a file's protection status. With PRO/DECnet, use this command to change a remote file's protection status.

File protection is described in your *Hard Disk System User's Guide*. You may need to protect files from other users of your workstation or from other users. If you transfer your files to other systems, you may need to use the SET PROTECTION command to change the protection of a particular file.

File protection is provided for remote RSX or P/OS files at four levels:

System—At the system level, protection controls file access to users who have system privileges. These are users who have UIC group numbers of 10 octal or less. The exact range of system group numbers is usually determined by the system manager of that system. UICs are User Identification Codes consisting of group numbers and member numbers. In other words, users are divided into groups and several members may be in the same group. The system group is a special group of people who need complete access to the system; usually this means operators, system managers, and system programmers.

Owner—At the owner level, protection controls file access to the user of a particular UIC, that is to one member of a group—the owner of the files in the UIC.

Group—At the group level, protection controls file access to the members of a particular group, that is, to all users, including the owner, who have the same group number in their UICs as the owner of the file.

World—At the world level, protection controls file access to all users on that system, including those in the first three categories.

You can specify whether each level can read, write, extend/execute, or delete the file. Read access is the right to examine, print, or copy a file or files on a volume. Write access is the right to modify a file or to write files on a volume. Extend access is the right to enlarge a file (for example, with the APPEND command). This access applies to RSX and P/OS systems. For VMS systems, execute access applies rather than extend. Execute access is the right to run

files that contain programs. Delete access is the right to delete a file or files on a volume.

The system default protection code is as follows:

```
(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)
```

The protection as described above may differ with remote systems other than P/OS or RSX, depending on their protection scheme implementations.

Format

```
SET PROTECTION = protection-code [/qualifier[,...]] filespec
```

Example

```
$ SET PROTECTION = (SYSTEM:RWED,OWNER:RWED,GROUP:R,WORLD:R)-
$-EVE::"JONES SECRET"MEMO.DOC <DO>
```

protection-code

is the protection code to be given to the files created during your current login session.

The code takes the following format:

```
((SYSTEM[:n]),[OWNER[:n]],[GROUP[:n]],[WORLD[:n]])
```

where:

SYSTEM	is the system level protection code,
OWNER	is the owner level protection code,
GROUP	is the group level protection code,
WORLD	is the world level protection code,

n is the type of access that is to be allowed. The value for n can be any or all of the following values, in any order:

R—Read allowed. The user may read, copy, print or type (scroll through) the file, and if it is a task, run it.

W—Write allowed. The user (or the user's application) can add new information to the file.

E—Extend allowed. The user, or the user's tasks, can change the amount of disk space allocated to the file.

D—Delete allowed. The user (or the user's application) can delete the file.

Only those categories that are specified are allowed; all others remain protected. If you enter one of the categories, but do not specify a value for n, no access is allowed at that level.

filespec

The file descriptor of the file(s) to be changed.

qualifier

May be one or more of the following:

/DATE:dd-mmm-yy

/SINCE:dd-mmm-yy

/THROUGH:dd-mmm-yy

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

/TODAY:dd-mmm-yy

/EXCLUDE:filespec

/NOWARNINGS

With PRO/DECnet, the following qualifiers are also available:

/LOG /OWNER:uic

Prompts

File(s)?

Protection Code?

Qualifiers

/DATE:DD-mmm-yy

The /DATE qualifier specifies that you wish the SET PROTECTION command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The /SINCE qualifier specifies that you wish the SET PROTECTION command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **SET PROTECTION** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **SET PROTECTION** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **SET PROTECTION** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **SET PROTECTION** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the "No such file" error messages should not be displayed, if the input files specified do not exist.

Additional Qualifiers Available with PRO/DECnet**/LOG**

The **/LOG** qualifier specifies that the names of files on which protection was changed are to be displayed as the operation is performed.

/OWNER:uic

Sets the ownership of a file to the specified UIC, in the form:
[group,member]

SET PROTECTION/DEFAULT

Establishes the default protection for all files created during the current login session. The protection for a file limits the type of access available to other system users.

Format

SET PROTECTION = protection-code/DEFAULT

Example

```
$ SET PROTECTION = (SYSTEM:RWE,OWNER:RWED,GROUP:R,WORLD:R-
$/DEFAULT <DO>
```

protection-code

is the protection code to be given to the files created during your current login session.

The code takes the following format:

```
([SYSTEM[:n]], [OWNER[:n]], [GROUP[:n]], [WORLD[:n]])
```

where:

SYSTEM	is the system level protection code,
OWNER	is the owner level protection code,
GROUP	is the group level protection code,
WORLD	is the world level protection code,

n is the type of access that is to be allowed. The value for n can be any or all of the following values, in any order:

R—Read allowed. The user may read, copy, print or type (scroll through) the file, and if it is a task, run it.

W—Write allowed. The user (or the user's application) can add new information to the file.

E—Extend allowed. The user, or the user's tasks, can change the amount of disk space allocated to the file.

D—Delete allowed. The user (or the user's application) can delete the file.

Only those categories that are specified are allowed; all others remain protected. If you enter one of the categories, but do not specify a value for *n*, no access is allowed at that level. In addition, if a category is not entered, then access is denied for that category. For example, SET DEFAULT PROTECTION = (OWNER:RWED) allows only owner access to the files created.

NOTE: Your default file protection is set during the login process. You may specify a default file protection that is restored during the login process with Environment Services.

SET PROTECTION/NODEFAULT

Removes your currently defined default file protection. (See SET PROTECTION/DEFAULT.) Any file created during the current login session will have the file protection of:

(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)

Format

SET PROTECTION/NODEFAULT

SET TERMINAL

This command sets various attributes of the terminal.

Format

```
SET TERMINAL[:ttn:] attribute[/attribute[s]]
```

ttn

The number of the terminal.

attribute

May be one or more of the following:

Group 1: Common Use Attributes

```
/[NO]LOWERCASE
/[NO]UPPERCASE
/WIDTH:n
```

Group 2: Terminal Setup Attributes

```
/LA100
/VT100
/VT101
/VT102
/VT105
/VT125
/VT131
/VT132
/VT2XX
/[NO]FORM_FEED
/[NO]HARDCOPY
/[NO]SCOPE
/[NO]TAB
/PAGE_LENGTH:n
```

Group 3: Task Setup Attributes

```
/[NO]ECHO
/[NO]EIGHT_BIT
/[NO]ESCAPE
/[NO]TYPEAHEAD[:n]
/[NO]WRAP
```

Prompts

Terminal attribute?

Attributes

You can set several attributes at once. If one of these commands fails, all others following it in the command line also fail.

Group 1: Common Use Attributes

The following parameters set terminal characteristics that are regularly needed by the average terminal user.

/[NO]UPPERCASE

/UPPERCASE is the default. All characters typed on the terminal appear in uppercase. **/NOLOWERCASE** is the same as **/UPPERCASE**.

/[NO]LOWERCASE

Characters typed on the terminal in lowercase appear in lowercase. Most system tasks will accept input in lowercase. Note that some terminal escape sequences use lowercase characters. If, for instance, the keypad commands in PRO/EDT do not seem to be working, you may have to set the **/LOWERCASE** attribute.

/WIDTH:n

The **SET TERMINAL/WIDTH** command sets the width of your terminal, that is, the length of a line. *n* can be from 0 through 132. Note that a line length of zero means no commands can be entered on the terminal.

Group 2: Terminal Setup Attributes

The following qualifiers set hardware characteristics of terminals. For certain common models, you can set a number of characteristics automatically simply by identifying the model. For other terminals, you must set these characteristics explicitly.

Here are the models for which setup is provided:

VT100	VT125
VT101	VT131
VT102	VT132
VT105	LA100
LA50	VT2XX

All of these are DIGITAL terminals. Those with “LA” designations are hard-copy terminals; those with “VT” designations are video models.

Setting a terminal to a particular model does not mean that the terminal will behave like that model. It means only that the operating system will treat the terminal as if it were that model. This feature is intended primarily to deceive tasks that expect a certain terminal model or to identify what a terminal is.

You can set hard-copy terminals as video terminals and you can set video terminals as hard-copy terminals. For the terminal user, the most noticeable difference will be in the way the DELETE key operates. Setting terminals from hard-copy to video may prove disruptive; setting terminals from video to hard-copy is less likely to cause trouble.

Setting a terminal to a particular model designation automatically sets a number of attributes for the terminal.

You can also set these attributes individually.

You can find out how your terminal has been identified by issuing a SHOW TERMINAL command.

/NOHARDCOPY
/SCOPE

Sets the terminal as a video terminal. /SCOPE is the equivalent of /NOHARDCOPY.

/HARDCOPY
/NOSCOPE

Sets the terminal as a hard-copy terminal. /NOSCOPE is the equivalent of /HARDCOPY.

/[NO]FORM__FEED

If the terminal hardware supports form feeds, the terminal should be set `/FORM__FEED`. If form feeds are handled by the software providing line feeds, the terminal should be set `/NOFORM__FEED`.

/[NO]TAB

If the terminal hardware supports horizontal tabs, the terminal should be set `TAB`. If tabs are handled by the software providing spaces, the terminal should be set `/NOTAB`.

/PAGE__LENGTH:n

Defines the number of lines to a page. By convention, a page is usually considered the number of lines to a screenful on video terminals, or the number of lines between perforations on hard-copy terminals. This attribute sets the length of the page.

Group 3: Task Setup Attributes

The following parameters set terminal characteristics that may be needed by system or user tasks. Most system tasks that require these attributes in a terminal will set the attributes when they attach the terminal. User tasks can also do this. This will be transparent to the user. These parameters are included for use in cases where the task does not do this setting.

/[NO]ECHO

Enables (or disables) echoing of characters typed on the terminal.

`/ECHO` is the default. Characters typed on the terminal are echoed on the terminal.

`/NOECHO` suppresses the echo.

`/NOECHO` changes nothing but echoing. Commands can still be passed to the system, but the system passes no echo back.

/[NO]ESCAPE

Enables (or disables) recognition of escape sequences from a terminal.

`/NOESCAPE` is the default. When you press the `ESC` key, (`CTRL/3` for the LK201 keyboard), it is interpreted as a line terminator (with a line feed but no carriage return).

ESCAPE enables the recognition of escape sequences from the terminal. When you press the ESC key, (CTRL/3 for the LK201 keyboard) it is interpreted as the beginning of an escape sequence and will no longer terminate a line.

This is a rarely used parameter. Most tasks that recognize escape sequences will attach the terminal so that escape sequences are passed without the user's knowledge, usually from the keypad.

`/[NO]WRAP`

Specifies that the terminal automatically wrap (or not wrap) lines longer than its line width.

`/WRAP` is the default. The terminal automatically issues a carriage return and a line feed when you type to whatever line width the terminal is set for.

`/NOWRAP` overrides this feature and permits unlimited line length.

`/[NO]EIGHT_BIT`

Enables (or disables) a terminal's `/EIGHT_BIT` attribute.

`/EIGHT_BIT` is the default, because the Professional 300 series uses eight-bit mode to display the DEC Multinational Character Set. `/EIGHT_BIT` allows the terminal to pass all eight bits of the ASCII character. This attribute is used when your terminal is communicating with some device that sends 8-bit ASCII.

SHOW CLOCK_QUEUE

SHOW CLOCK_QUEUE displays information about tasks that are currently in the clock queue. The displayed information comprises task names, the next time the task is scheduled to run, and each task's reschedule interval, if any.

Format

```
SHOW CLOCK_QUEUE
```

Prompts

None

SHOW COMMON

SHOW COMMON displays the names of resident commons installed in the system, their PCB addresses, the number of attached tasks, and the status of each common.

You can also display information about a single common optionally, including a list of tasks attached to the common.

If you do not name a common, information about all commons in the Common Block Directory is displayed.

Format

```
SHOW COMMON[:name][/qualifier]
```

qualifier

```
/TASK
```

name

The name of the common.

The display is in the following format:

```
common pcbaddr taskcount statusbits
```

where:

commonname	is the name of the common.
pcbaddr	is the address of the Partition Control Block (PCB).
taskcount	is the number of tasks mapped to the common.
statusbits	is a list of common region status bits that are set.

The following is a list of status bits and their meanings:

Status Bit	Meaning
CAF	Checkpoint allocation failure
-CHK	Not checkpointable
CKP	Checkpoint in progress
CKR	Checkpoint requested
COM	Library or common
DEL	Marked for delete on last detach
DRV	Driver common
FXD	Fixed in memory
LIO	Long I/O
LFR	Last load failed
NSF	Not shufflable
OUT	Out of memory
PER	Parity error
PIC	Position independent
RON	Read-only common

Usually, the status bits will indicate that the common region is either fixed in memory (FXD) or out of memory (OUT).

Prompts

None

Qualifier

/TASK

Specifies that you wish a list of tasks attached to a particular common region showing the number of times each task is mapped to the common (mapping count).

SHOW [DAY]TIME

This command displays the current day and time.

Format

```
SHOW [DAY]TIME
```

Prompts

None

SHOW DEFAULT

This command displays the current default device and directory name.

Format

SHOW DEFAULT

Prompts

None

SHOW DEVICES

This command displays information concerning the devices that are included in the system.

Format

SHOW DEVICES[qualifier]

qualifier

May be one or more of the following:

/dd[nn]:

/WIDTH:ddnn:

Prompts

None

Qualifiers

If you do not include a qualifier, **SHOW DEVICES** displays a list of all the devices on the system, including terminals and pseudo devices.

/dd[nn]:

Displays information about all devices of a particular type on the system. Where dd: is the two-letter device mnemonic terminated with a colon, which indicates the type of device controller. The display shows the devices under that type of controller by model name.

/WIDTH:ddnn:

Displays the size of the I/O buffer (line length) for a particular device, including a terminal.

SHOW LOGICALS or SHOW ASSIGNMENTS

SHOW LOGICALS displays all the logical names in the logical name table(s), or it displays the current equivalence name assigned to a specified logical name.

Format

```
SHOW LOGICALS [logicalname] [/qualifier(s)]  
SHOW ASSIGNMENTS [logicalname] [/qualifier(s)]
```

logicalname

The name of a unique logical assignment. This can also be a quoted logical name.

Qualifiers

/USER

Limits the display to the user logical name table. The user logical name table contains temporary logicals that are deleted by the system when you exit the DCL application.

/SESSION

Limits the display to the session logical name table. The session logical name table contains logical names that will remain in the system until you logout.

/SYSTEM

Limits the display to the system logical name table. The system logical name table contains logicals that everyone logged into the system have access to. System logicals are removed either by using the DEASSIGN command, or when the system is powered down.

/ALL

Displays the entire contents of the logical name tables.

Prompts

None

SHOW PROTECTION

SHOW PROTECTION displays the current file protection to be applied to all new files created during the current session. You can change the default protection at any time using the **SET PROTECTION/DEFAULT** command or the P/OS Environment Services **SET DEFAULT FILE PROTECTION** option.

Format

SHOW PROTECTION

Prompts

None

SHOW TASKS/ACTIVE

SHOW TASKS/ACTIVE displays information about active tasks in brief and full format.

Brief Format

SHOW TASKS/ACTIVE[:ttn:][[/qualifier[s]]]

ttn:

The name of the terminal you want information about.

If you name a terminal in the command, the display shows in brief form the tasks active at that terminal. If you do not name a terminal, the display shows in brief form the tasks active at your terminal.

qualifiers

May be one or more of the following:

/BRIEF

/ALL

Prompts

None

Qualifiers

/BRIEF

Specifies that you wish to display information about active tasks in the brief format. This is the default and need not be specified.

The brief format includes task names and the originating terminal in parentheses next to each task name.

/ALL

Specifies that you wish to display information about all tasks active on the system. The default is to show information about tasks active at your terminal only.

Full Format**SHOW TASKS/ACTIVE/FULL** [taskname]

or

SHOW TASK[:taskname]/ACTIVE/FULL

taskname

The name of the task you want information about.

If you include a task name, the display shows full information on that task. If you do not name a task, the display shows full information on all currently active tasks.

Qualifiers**/FULL**

Requests the full format display for the **SHOW TASK/ACTIVE** command. This format includes detailed information on the state of one or all tasks active on the system.

The following display shows the state of the task at the time the command was issued.

The format of the display is as follows:

```
taskname tcbaddr parname pcbaddr taskaddrlimits pri defpri
STATUS: statusflags
TI - ttnn: IOC - iocount BOC - buffiount EFLG - eventflags PS - pswval
PC - pcval REGS 0-6 r1val r2val r3val r4val r5val r6val
```

where:

tcbaddr	is the physical address of the Task Control Block (TCB).
parname	is the name of the task's partition.
pcbaddr	is the physical address of the Partition Control Block (PCB).
taskaddrlimits	is the base and top of the task's dynamic subpartition as physical addresses.
pri	is the priority at which the task is actually running.

defpri	is the default priority at which the task was built or installed to run.
STATUS	is the task's status flags. These are identified in Table 3-2.
TI	is the initiating terminal.
IOC	is the decimal I/O count for the task.
BIO	is the decimal count of I/O buffered by the terminal driver and Executive.
EFLG	are the local event flags.
pswval	is the Processor Status Word.
pcval	is the Program Counter.
REGS	is the contents of the task's other registers. If the task was spawned by another task, the name of the parent task is also displayed.

If the task is not in memory (OUT flag displayed, see Table 3-2), the PC, PSW, and other registers are not displayed.

Table 3-2
Task Status Flags

<i>Status Flag</i>	<i>TCB Flag</i>	<i>Meaning</i>
ABO	T2.ABO	Being aborted
ACP	T3.ACP	Ancillary Control Processor
AST	T2.AST	AST state
BLK	TS.STP	Blocked externally by CLI command
CAF	T2.CAF	Dynamic checkpointing allocation failure
CAL	T3.CAP	Checkpoint space allocated in task image
-CHK	TS.CHK	Not checkpointable
CIP	TS.CIP	Blocked for checkpoint in progress
CKD	TS.CKD	Checkpointing disabled
CKP	TS.CKP	Checkpointed
CKR	TS.CKR	Checkpoint request pending
CLI	T3.CLI	Command Line Interpreter
DSP	T4.DSP	Task was built for user-mode I- and D-space
DST	T2.DST	ASTs disabled
-EXE	TS.EXE	Not executing
FXD	T2.FXD	Fixed in memory
GFL	T3.GFL	Task has own group global event flags locked

Table 3-2
Task Status Flags (continued)

Status Flag	TCB Flag	Meaning
LDD	T4.LDD	Task's load device has been dismantled
MUT	T4.MUT	Task is multiuser task
OUT	TS.OUT	Out of memory
-PMD	T3.PMD	Suppress PMD on SST abort
PRO	T4.PRO	Task is (or should be) a prototype task
PRV	T3.PRIV	Privileged
PRV	T4.PRIV	Task was privileged but has cleared T3.PRIV
RDN	T3.RDN	I/O being run down
REM	T3.REM	Remove on exit
REX	T2.REX	Abort AST effected or in progress
ROV	T3.ROV	Resident overlays
RST	T3.RST	Restricted—used by layered software
SEF	T2.SEF	Stopped for event flag
SLV	T3.SLV	Slaved
SNC	T4.SNC	Task uses commons for synchronization
SPN	T2.SPN	Being suspended
SPNA	T2.SPN	Suspended prior to AST
STP	T2.STP	Stopped
STPA	T2.STP	Stopped prior to AST
SWS	T3.SWS	Reserved for software services
WFR	T2.WFR	In a wait-for state
WFRA	T2.WFR	In a wait-for state prior to AST

NOTE: These status flags are displayed by several forms of the SHOW TASK command. They give information on what is happening within the task and between the task and the system. They also identify certain kinds of tasks. Names prefixed by a minus (-) indicate the complement of the condition. Thus, -CHK means the task is noncheckpointable. You must understand how the system runs tasks to understand the meanings of all these flags. See the *RSX-11M/M-PLUS Task Builder Manual* and the *P/OS System Reference Manual* for more information.

SHOW TASKS/INSTALLED

SHOW TASKS/INSTALLED displays information about installed tasks in either brief or full format.

Format

SHOW TASKS[:taskname]/INSTALLED[/qualifier]

taskname

Specifies the task for which you wish information displayed. If you do not specify a task name, information on all installed tasks is displayed.

qualifiers

May be one or more of the following:

/BRIEF

/FULL

Prompts

None

Qualifiers

/BRIEF

Requests information on installed tasks in a brief format. This is the default and need not be specified. The format of the display is as follows:

taskname ident parname priority size ddnn:-lbn [memstate]

where:

taskname	is the name of the task.
ident	is the task version identification (or the version of the prototype task).
parname	is the partition in which the task is installed.
priority	is its priority.
size	is the size of the task in bytes.
ddnn:	is the device from which it is to be loaded.

lbn is the logical block number of its disk address.

memstate is the task memory state, which can be fixed, checkpointed, or blank.

If the task version identification is missing (with the rest of the line moved left) or if it is garbage, the task was installed from a disk that is no longer present. If the task version number is a date, such as 07JUL, the task was compiled on that day.

/FULL

Requests the full format of the SHOW TASKS/INSTALLED command. This format displays a detailed list of the states of one or all of the installed tasks in the system, depending on the presence of the taskname parameter. The format of the display is as follows:

```
taskname tcbaddr parname pcbaddr taskaddrlimits pri defpri
STATUS: statusflags
TI - ddn: IOC - iocount EFLG - eventflags
```

This display is in the same format as that of SHOW TASKS/ACTIVE/FULL. (See the SHOW TASKS/ACTIVE/FULL command for a description of the display.)

SHOW TASK/LOGICAL_UNITS

SHOW TASK:taskname/LOGICAL_UNITS displays the static logical-unit-number assignments (LUNs) for an installed task.

Format

SHOW TASK:taskname/LOGICAL_UNITS

taskname

Specifies the task for which you want LUN assignments displayed. This must be the name of a task installed through INSTALL, and not through the install-run-remove form of RUN.

The display consists of a list of physical devices and corresponding static LUN assignments. The display does not show any dynamic LUN assignments, even when the specified task is running.

Prompts

None

SHOW TERMINAL

SHOW TERMINAL displays information about your terminal and another optional terminal connected to the system.

Format

```
SHOW TERMINAL[:ttn:] [/attributes[s]]
```

:ttn:

The number of the terminal for display.

attribute

May be one or more of the following:

Group 1: Common Use Attributes

```
/[NO]LOWERCASE  
/[NO]UPPERCASE  
/WIDTH:n
```

Group 2: Terminal Setup Attributes

```
/LA100  
/VT100  
/VT101  
/VT102  
/VT105  
/VT125  
/VT131  
/VT132  
/VT2XX  
/[NO]FORM_FEED  
/[NO]HARDCOPY  
/[NO]SCOPE  
/[NO]TAB  
/PAGE_LENGTH:n
```

Group 3: Task Setup Attributes

`/[NO]ECHO`
`/[NO]EIGHT_BIT`
`/[NO]ESCAPE`
`/[NO]TYPEAHEAD[:n]`
`/[NO]WRAP`

Prompts

None

Attributes

Each `SHOW TERMINAL` attribute is directly related to a `SET TERMINAL` attribute (refer to the `SET TERMINAL` command for a discussion of these attributes).

SPAWN

The SPAWN command allows commands to be executed in the background. These tasks allow you to continue to use the terminal. For example, SPAWN can initiate a task building session so you can then perform other tasks, such as edit a file using the terminal.

SPAWN should not be used with commands or tasks that attach the terminal, i.e. read or write from or to the terminal. SPAWN will return an error message, if it is asked to initiate an uninstalled task.

Format

SPAWN DCL-COMMAND

DCL-COMMAND

Any Command Language command.

Prompts

None

START

START resumes the execution of a task that was halted by a **STOP\$\$** directive. Starting a task that has been stopped is different from continuing a suspended task.

Format

START taskname

taskname

The name of the task you wish to start.

Prompt

Task name?



START/UNBLOCK

This command continues the execution of a task that was blocked by the STOP/BLOCK command.

Format

START/UNBLOCK taskname

taskname

The name of the task to be unblocked.

Prompts

None

STOP/BLOCK

This command blocks an installed, active task. After execution of this command, the task no longer executes or competes for memory (refer to START/UNBLOCK).



Format

STOP/BLOCK taskname

taskname

The name of the task to be blocked.

Prompt

Task name?



SUBMIT/REMOTE

With PRO/DECnet only, SUBMIT/REMOTE directs an existing command file on a remote node to be executed. The file is queued to the remote node's command file or batch file processor. If the command file is not on the node where it will be executed, you may wish to use a COPY command with a /SUBMIT qualifier.

The success of the command does not guarantee that the batch or command file was executed successfully, only that the execution request was given successfully to the remote processor.

Format

```
SUBMIT/REMOTE[/qualifier[...]] commandfile[...]
```

Example

```
$ SUBMIT/REMOTE SMAUG::[DOOLEY]CLEANUP.COM <DO>
```

qualifier

May be one or more of the following:

```
/LOG
/NOWARNINGS
```

commandfile

The output file descriptor for each remote node command file that you want to execute. Command files are not deleted after execution.

Prompt

```
File(s)?
```

Qualifiers

/LOG

Specifies that the names of files submitted are to be displayed as the operation is performed.

/NOWARNINGS

Specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

TYPE

This command displays the contents of a file or group of files on the screen.

With PRO/DECnet, you can display the contents of remote files. The file displays are transferred to your Professional in ASCII record mode.

Format

```
TYPE[/qualifier[,...]] filespec[,...]
```

Example

```
$ TYPE MYFILE.TXT <DO>
```

qualifier

May be one or more of the following:

```
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
/SHARED
```

PRO/DECnet allows the following additional qualifiers:


```
/LOG
/MACY11
```

filespec

The name of the file to be displayed.

Prompt

```
File(s)?
```

Qualifiers **/DATE:dd-mmm-yy**

The **/DATE** qualifier specifies that you wish the **TYPE** command to affect only files created on the designated date.

/SINCE:dd-mmm-yy

The **/SINCE** qualifier specifies that you wish the **TYPE** command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **TYPE** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **TYPE** command to affect only files created within the range of designated dates.

 **/TODAY**

The **/TODAY** qualifier specifies that you wish the **TYPE** command to affect only files created on the same day as the command is issued.

/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **TYPE** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the “*” wildcard.

/NOWARNINGS

Specifies that the “No such file” error messages should not be displayed when the input files specified do not exist.

 **/SHARED**

Specifies that you want other users to be able to access the file while you are typing it.

Additional Qualifiers Available with PRO/DECnet

/LOG

Specifies that the names of files typed and their sizes are to be displayed as the operation is performed.

/MACY11

Specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACY11 format. This qualifier is used to inform the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACY11 must be entered after the file specification in the command line.

UNLOCK

UNLOCK attempts to make available to the system one or more locked files. Locked files are files that have been improperly closed because a task aborted or stopped execution while the file was open. Locked files are identified by an "L" in the directory listing.

Format

```
UNLOCK[/qualifier[,...]] filespec[,...]
```

Example

```
$ UNLOCK PAYROLL.DAT <DO>
```

qualifier

May be one or more of the following:

```
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
```

filespec

The name of the file to be unlocked.

Prompt

File(s)?

Qualifiers

```
/DATE:dd-mmm-yy
```

The /DATE qualifier specifies that you wish the UNLOCK command to affect only files created on the designated date.

```
/SINCE:dd-mmm-yy
```

The /SINCE qualifier specifies that you wish the UNLOCK command to affect only files created on or after the designated date.

/THROUGH:dd-mmm-yy

The **/THROUGH** qualifier specifies that you wish the **UNLOCK** command to affect only files created on or before the designated date.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy

The **/SINCE** and **/THROUGH** qualifiers can be combined to specify that you wish the **UNLOCK** command to affect only files created within the range of designated dates.

/TODAY

The **/TODAY** qualifier specifies that you wish the **UNLOCK** command to affect only files created on the same day as the command is issued.

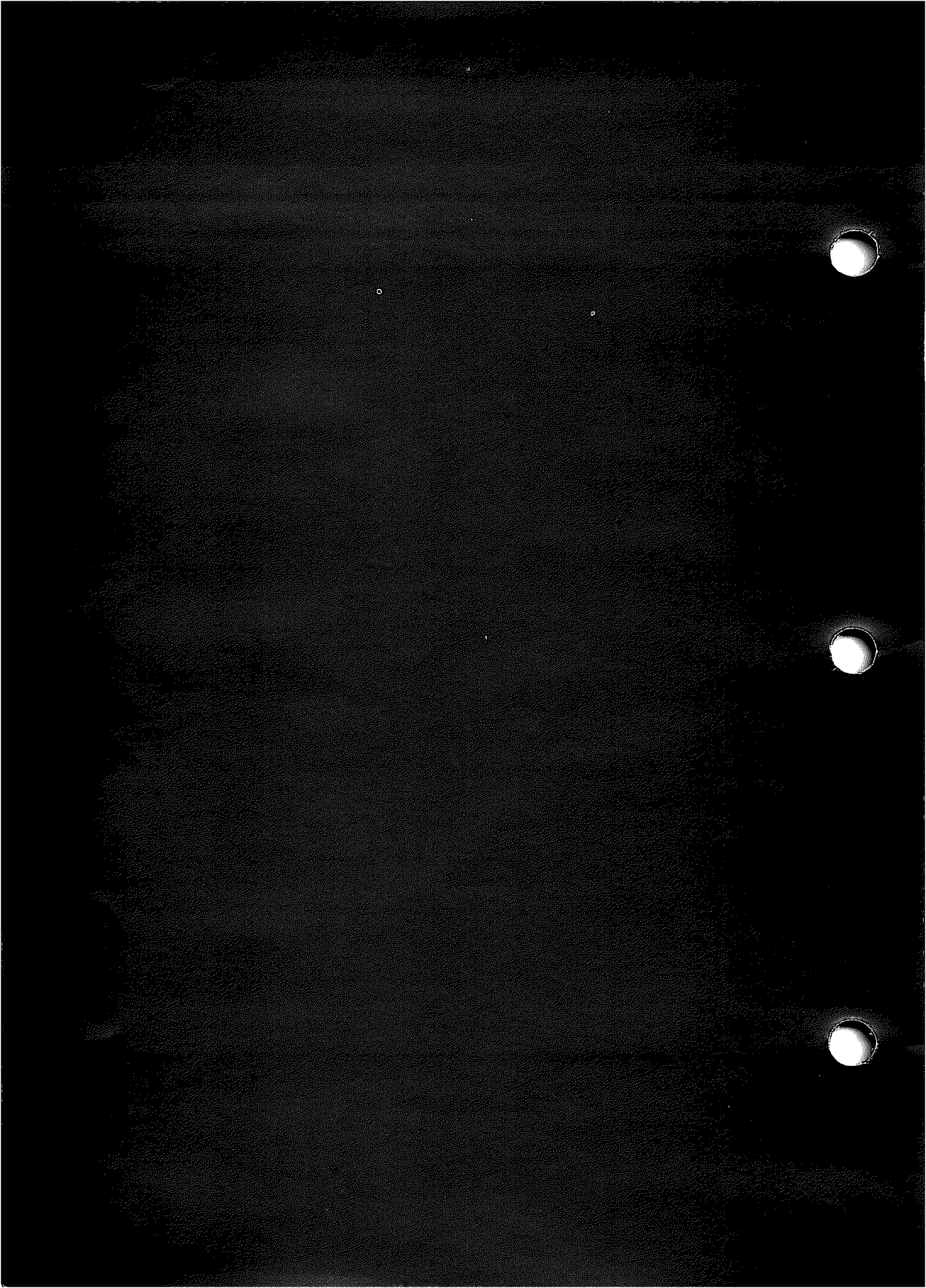
/EXCLUDE:filespec

The **/EXCLUDE** qualifier specifies that you wish the **UNLOCK** command not to affect certain files. The **filespec** argument to **/EXCLUDE** can contain wildcards, but the file specification must contain a version number, either explicitly or as the "*" wildcard.

/NOWARNINGS

The **/NOWARNINGS** qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

Appendix A



Appendix A

Common Error Messages.

The following error messages are common to many DCL commands. They may appear on your screen preceded by a three-letter code identifying the system component that detected the error. This can be DCL or the first three letters of the command itself. This code appears here as yyy.

Most of the error messages with the word “expected” in them reprint the command on your screen with a circumflex (^) pointing to the error. Sometimes the circumflex points to the character just past the last successfully parsed command element.

Many system programs (tasks) other than DCL also issue error messages when an error is made while using a DCL command. Some error messages may be difficult to understand if you have not used other versions of DCL. Make sure you have used the correct command format (see Chapter 3). Retry the command. If you still get an error code, write it down along with the circumstances under which the error occurred and seek help from your Professional support person.

The following is a list of common error messages for DCL commands.

yyy–Allocation failure–no contiguous space

Explanation: Not enough contiguous space is available on the output volume for the file being copied.

User Action: Delete any files no longer required on the output volume and retry the command.

yyy–Allocation failure–no space available

or

yyy–Allocation failure on output file

Explanation: Not enough space is available on the output volume for the file being copied.

User Action: Delete any files no longer required on the output volume and retry the command.

yyy–A–Z and/or 0–9 expected

Explanation: The command as typed included a nonalphanumeric character.

User Action: Check the command for the proper format and reenter.

yyy–A–Z expected

Explanation: The command as typed included a nonalphabetic character.

User Action: Check the command for the proper format and reenter.

yyy–Bad error message

Explanation: Some unusual condition has caused an error.

User Action: Record the command that caused the error and other information on activity at your screen at the time and seek help from your Professional or application support person.

yyy–Bad use of wildcards in destination file name

Explanation: A wildcard (*) was specified for an output file where it is not permitted.

User Action: Reenter the command with a complete and explicit file specification for the output file.

yyy–Cannot find directory file

Explanation: The command specified a directory not found on the current volume.

User Action: Reenter the command after checking for the correct directory and correct volume.

yyy-Cannot find files

Explanation: The file or files specified in the command are not in the designated directory.

User Action: Check the file specification and reenter the command line.

yyy-Cannot rename from one device to another

Explanation: The command attempted to rename a file across devices (volumes).

User Action: Use the COPY command to move the file from one volume to another and rename it.

yyy-Close failure on input file

or

yyy-Close failure on output file

Explanation: A file named in the command could not be properly closed. The file is locked.

User Action: Use UNLOCK to unlock the file. Determine the cause of the error and correct it if you can. Sometimes a file cannot be unlocked, or, if unlocked, it may be altered.

yyy-Command function not unique

Explanation: The command as typed did not include sufficient characters to identify some command function.

User Action: Retype the command after checking the proper format.

yyy-Command line incomplete

Explanation: The command as typed is not a complete command.

User Action: Retype the command after checking the proper format.

yyy-Command syntax error

Explanation: The command did not conform to the format rules.

User Action: Check the command for the proper format and reenter it.

yyy-Conflicting qualifier

Explanation: The command as typed included qualifiers that conflict with each other in their effect.

User Action: Retype the command after checking the proper format.

yyy-Contradictory qualifier

Explanation: The command as typed included contradictory qualifiers, such as /LOG and /NOLOG.

User Action: Check for the proper format and reenter the command.

yyy-Device invalid or not specified

Explanation: The command specified an invalid device or no device at all when a device name is required.

User Action: Check the devices on the system (View Status). Reenter the command after checking for the proper format. Determine the cause of the error and correct it if you can.

yyy-Device not in system

Explanation: The command specified a device that is not in the current system.

User Action: Make sure you have inserted any diskette volume you are trying to use.

yyy-Device not mounted/allocated

Explanation: The command specified a device that is not properly mounted or allocated for the command to execute.

User Action: Make sure you have inserted any diskette volume you are trying to use.

yyy-Device not terminal

Explanation: The command specified a device other than a terminal where a terminal device name is required. TI: is not an acceptable terminal device name in all contexts.

User Action: Reenter the command after checking for the proper format.

yyy-Directory write-protected

Explanation: The command attempted to remove an entry from a directory that is privileged or from a directory on a device that is write-protected.

User Action: You may have typed an incorrect directory name. Some directories are protected from change. Diskettes can be write-protected. Check to see what applies to your situation.

yyy-Error dispatching command-DSW = 'n'

Explanation: An error occurred that was not explicitly handled by DCL or some invoked task.

User Action: Write down the error message and the circumstances under which it occurred. Then consult your Professional or application support person.

yyy-Explicit output file name required

Explanation: The command requires an explicit output file name.

User Action: Reenter the command line in the proper format (without wildcards).

yyy-Extraneous input

Explanation: The command as typed included extraneous input. The circumflex (^) points to the error or just past the last successfully completed command element.

User Action: Reenter the command line in the proper format.

yyy-Failed to attach output device

or

yyy-Failed to detach output device

Explanation: An attempt to attach or detach a record-oriented output device, such as a terminal or printer, failed. This error usually means the device is off-line (turned off) or nonresident (not inserted).

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your Professional or application support person.

yyy-Failed to create output directory

Explanation: The command failed to create an entry in a directory because the device was write-protected or because of a privilege violation.

User Action: Determine the cause of the error and correct it if you can. If you cannot, consult your Professional or application support person.

yyy-Failed to delete file

or

yyy-Failed to mark file for delete

Explanation: The command attempted to delete a protected file.

User Action: Check for the proper file specification and default directory and reenter the command. You should not delete system files. Refer to your User's Guide for additional information about changing file protection.

yyy-Failed to enter new filename

Explanation: The command specified a file that already exists in the directory, or the directory is protected.

User Action: Check for the proper format and reenter the command.

yyy-Failed to find files

Explanation: The command specified a file or files that could not be found as specified.

User Action: Check for the proper file specifications and reenter the command.

yyy-Failed to get time parameters

Explanation: An internal system problem has occurred.

User Action: Retry the command. If the error recurs, consult your Professional or application support person.

yyy-Failed to read attributes

Explanation: The command specified a volume that is either corrupted or protected against access.

User Action: You may be able to correct the error by making your defaults the same as the device and directory of the file you wish to affect.

yyy-Failed to remove directory entry

Explanation: The command attempted to remove an entry from a directory or volume that was protected against change.

User Action: Determine the cause of the error and correct it if you can. You should not change files in system directories. If you are trying to delete a file from a diskette, make sure it is not write-protected. Consult your User's Guide for more information about changing protection.

yyy-Failed to write attributes

Explanation: The command specified a volume that is corrupted or is protected against access.

User Action: You may be able to correct the error by making your defaults the same as the device and directory of the file you wish to affect.

yyy-Fatal I/O error

Explanation: The command failed to execute because of some I/O (Input/Output) error. This error can be caused by the unavailability of a device or by a device error. The device may be write-locked.

User Action: Determine the cause of the error and correct it if you can. If you cannot, consult your Professional or application support person.

yyy-File is lost

Explanation: P/OS has removed a file from its directory, failed to delete it, and failed to restore the directory entry.

User Action: Consult your Professional or application support person.

yyy-File not locked

Explanation: An UNLOCK command specified a file that is not locked.

User Action: Check for the proper file specification and reenter the command.

yyy-Filename or filetype not specified

Explanation: The command as typed did not clearly specify a file name and file type where one or the other or both is required. This error can be caused if you do not leave a space in front of a file specification that is a parameter.

User Action: Retype the command after checking for the proper format.

yyy-File specification either invalid or not specified

Explanation: The system could not read a file specification included in the command. This error often results from a typing mistake or typing the command in the wrong format.

User Action: Check for the proper format and reenter the command.

yyy-File version number not specified

Explanation: The command requires a file version number to be specified.

User Action: Reenter the command after checking for the proper format.

yyy-Function not unique

Explanation: The command as typed did not include sufficient characters to identify some function.

User Action: Retype the command but include more characters.

yyy-Get command line-bad FILENAME

Explanation: The command specified an illegal indirect command file.

User Action: Check for the proper file specifications and reenter the command.

yyy-Illegal command

Explanation: The command, which is the first word on the command line, is not part of DCL.

User Action: Check the command for the proper format and reenter.

yyy-Illegal device

Explanation: The command named a device in an illegal format or contained some other format error.

User Action: Check the command for the proper format and reenter. Device names are two alphabetical characters followed by an octal number and a colon, or the names listed in Chapter 2 under File Specification-device.

yyy-Illegal filespec

Explanation: The command required a file specification that was not present.

User Action: Check the command for the proper format and reenter. See Chapter 2 for a complete description of a file specification. Perhaps some other command element is being used as a filespec. Use the prompting version.

yyy-Illegal or contradictory qualifier

Explanation: One or more qualifiers to the command are in conflict, or are in error.

User Action: Check the command for the proper format and reenter.

yyy-Illegal qualifier value

Explanation: The command as typed included an improper argument to a qualifier.

User Action: Check the command for the proper format and reenter. In DCL, an argument is preceded by a colon (:).

yyy-Illegal task name

Explanation: The command named a task using a task name in an illegal format.

User Action: Check the command for the proper format and reenter. Task names include as many as six Radix-50 characters. The Radix-50 characters are the uppercase alphabet, the numbers 0 through 9, the dollar sign (\$), and the period (.).

yyy-Illegal use of wildcard character

Explanation: The command included a wildcard (* or %) in a file specification in a way that would result in unpredictable or inconsistent output.

User Action: Check the command for the proper format and reenter. You may not be able to use the wildcard.

yyy-Input device must be a directory device

Explanation: The command as typed specified a device that is not a directory device, such as a printer. Directory devices are hard disk and diskettes.

User Action: Correct the format and reenter the command.

yyy-Input files have conflicting attributes

Explanation: Warning message. The command operation was completed, but the files named had conflicting attributes.

User Action: If you know this is a problem, consult your Professional or application support person.

yyy-Invalid command function

Explanation: The command as typed requested a function that is not valid for that command.

User Action: Check the command for the proper format and reenter it.

yyy-Invalid command parameter

Explanation: The command as typed included a parameter that is not valid. In DCL, a parameter is either entered in response to a prompt or preceded by a space.

User Action: Check the command for the proper format and reenter.

yyy-Invalid file specification list

Explanation: The command included a list of file specifications in an invalid format. In general, file specifications in lists should be separated by commas and, optionally, blanks.

User Action: Retype the command after checking the proper format.

yyy-Invalid file specification qualifier .

Explanation: The command included an invalid qualifier to a file specification. In DCL, a qualifier is preceded by a slash (/).

User Action: Check for the proper format and retype the command.

yyy-Invalid time or date

Explanation: The command specified a clock or calendar field, or both, incorrectly.

User Action: Check for the proper format and retype the command.

yyy-Invalid UIC specified

or

yyy-Invalid directory specified

Explanation: The command specified a directory name (UIC) that was a nonalphanumeric string or more than nine characters.

User Action: Retype the command after checking for the proper format.

yyy-I/O error on input file

or

yyy-I/O error on output file

Explanation: One of the following conditions exists:

- The device is not mounted (inserted).
- The hardware has failed.
- The output volume is full.
- The input file is corrupted.

User Action: Determine the cause of the error and correct it if you can. If you cannot, consult your Owner's Manual (hardware failure) or your Professional support person.

yyy–More command parameters than permitted

Explanation: The command as typed included too many parameters.

User Action: Check the command for the proper format and reenter.

yyy–No such file

Explanation: The command requested operations on a file that does not exist.

User Action: Make sure you have named the file properly. Check your defaults to be sure you are looking in the right directory on the right device. You may have made a typing error. If the desired file is in fact not present, find out why it is not present and proceed accordingly.

yyy–Not a directory device

Explanation: A directory-oriented command named a device that does not have directories, such as a printer.

User Action: If you can, reenter the command without specifying a directory.

yyy–Numeral expected

or

yyy–Numeral required

Explanation: The command included nonnumeric characters in a position where numerals are required or expected.

User Action: Check the command for the proper format and reenter.

yyy–Octal number expected

Explanation: The command included a number with an 8 or 9 where an octal number was expected.

User Action: Check the command for the proper format and reenter.

yyy–Open failure on file

Explanation: The system could not open a required file for some reason.

User Action: Check the directory to be sure that the file is present, not locked, and in the proper format. If the file is locked, it may be corrupted or contain bad data. Determine the cause of the error and correct it if you can.

yyy-Output device must be a directory device

Explanation: The command as typed specified a device that is not a directory device, such as a printer. Directory devices are hard disk and diskettes.

User Action: Correct the format and reenter the command.

yyy-Qualifier not available for this command format

Explanation: The command included a qualifier that is invalid in the current context of the command. The context is usually determined by some other qualifier in the command.

User Action: Reenter the command after checking for the proper format.

yyy-Qualifier not unique

Explanation: The command as typed did not include sufficient characters to identify some qualifier. In DCL, qualifiers are preceded by a slash (/).

User Action: Retype the command, but include more characters.

yyy-Qualifier value invalid here

Explanation: The command as typed included an inappropriate argument to a qualifier. In DCL, an argument is usually preceded by a colon (:).

User Action: Check the command for the proper format and reenter.

yyy-Radix-50 expected

Explanation: The command included a non-Radix-50 character where Radix-50 was required.

User Action: Reenter the command using the proper format. The Radix-50 characters are the uppercase alphabet, the numbers 0 through 9, the dollar sign (\$), and the period (.).

yyy-Repeated command parameter

Explanation: The command as typed included one parameter more than once. In DCL, a parameter is preceded by a blank or prompt.

User Action: Retype the command after checking for the proper format.

yyy-Repeated qualifier

Explanation: The command specified the same qualifier more than once. In DCL, a qualifier is preceded by a slash (/).

User Action: Reenter the command after checking the format to see if it is right.

yyy-Required parameter not specified

Explanation: The command cannot execute without required parameters, such as a file specification or attribute. In DCL, a parameter is preceded by a blank or prompt.

User Action: Retry the command, using the prompts. DCL prompts for all required parameters.

yyy-Required qualifier not specified

Explanation: The command requires a qualifier that it does not include. In DCL, a qualifier is preceded by a slash (/).

User Action: Check for the proper format and reenter the command.

yyy-Required qualifier value not specified

Explanation: A qualifier to the command requires that you state a numerical argument. In DCL, an argument is usually preceded by a colon (:).

User Action: Retype the command after checking for the proper format.

yyy-Sorry, line too long

Explanation: DCL commands are translated for execution by P/OS or another system task or utility. This error is caused by a translated command line that the destination task cannot handle.

User Action: Check the command format to see if you are specifying elements that can be defaulted. Check to see if you can enter the command twice with different qualifiers to accomplish in two commands what you cannot accomplish in one.

yyy-Sorry, low pool

Explanation: The command could not execute because of insufficient space in the system pool (dynamic storage region).

The pool is the P/OS Executive's data base. In general, each task, including commands, uses a certain amount of pool. If the pool is full or badly fragmented, there may not be sufficient space for the command to execute. Usually, pool problems clear up spontaneously if you wait.

User Action: Retry the command after an interval. You should not attempt to use the **INTERRUPT DO** keys or run another application when the pool is low. These also use pool space.

yyy-Sorry, task active

Explanation: The command required some action that cannot be taken on an active task.

User Action: Determine the cause of the error and correct it if you can.

yyy-Sorry, task not installed

Explanation: The command attempted to invoke a task that was not installed.

User Action: Try running the task with a **RUN** command. Consult any relevant application documentation.

yyy-Syntax error

Explanation: The command included some error in typing or specification, such as a letter where a number should appear.

User Action: Check for the proper format and reenter the command.

yyy-Version must be explicit or "*"

Explanation: The command format requires that the version number of the file must be specified explicitly or as a "*" wildcard.

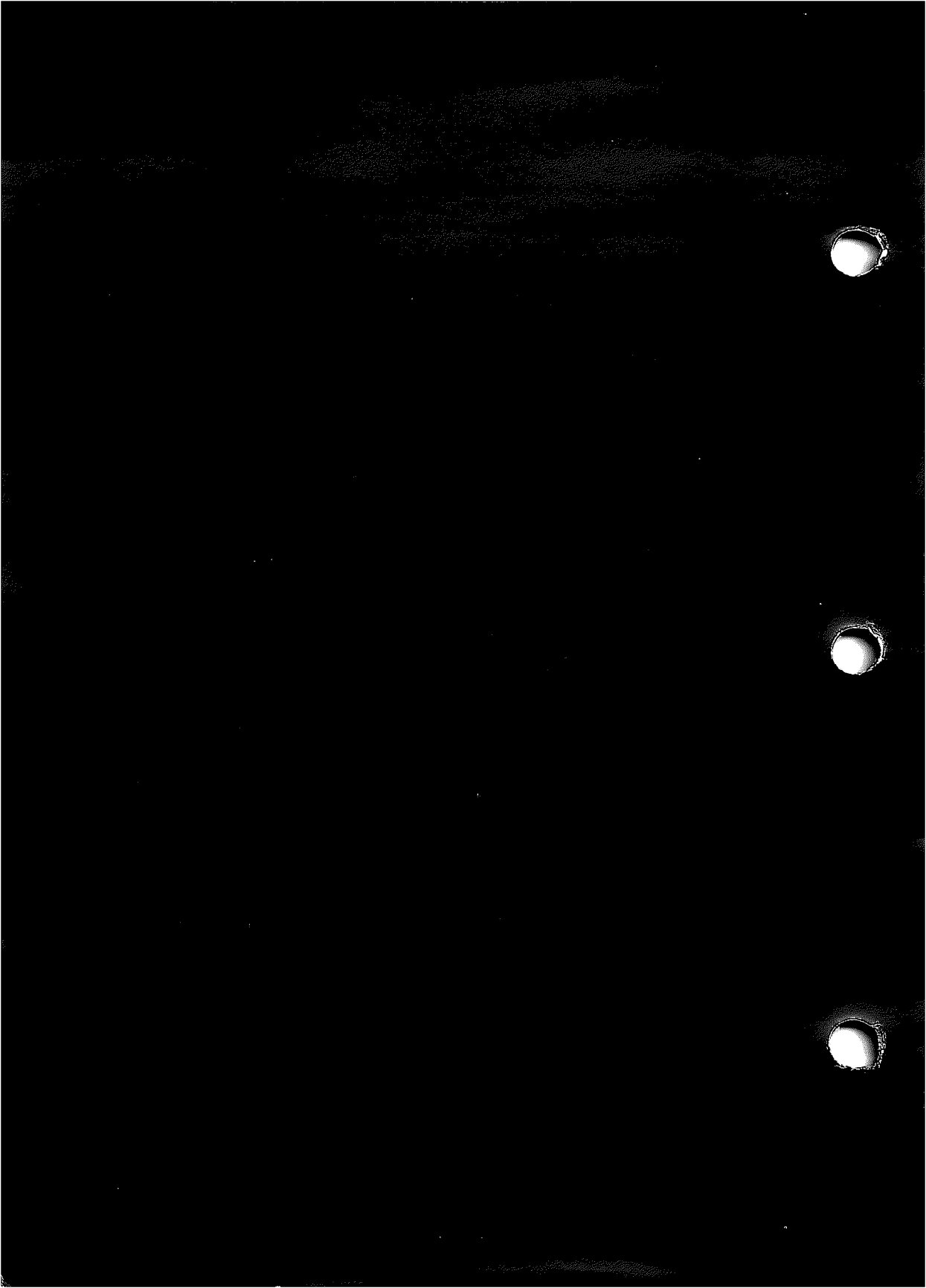
User Action: Reenter the command with the version number correctly expressed.

yyy-Wildcards not permitted

Explanation: The command included a wildcard (* or %) in a context where it is not permitted.

User Action: Check for the proper format and reenter the command.

Appendix B



Appendix B

Installing the PRO/EDT Editor

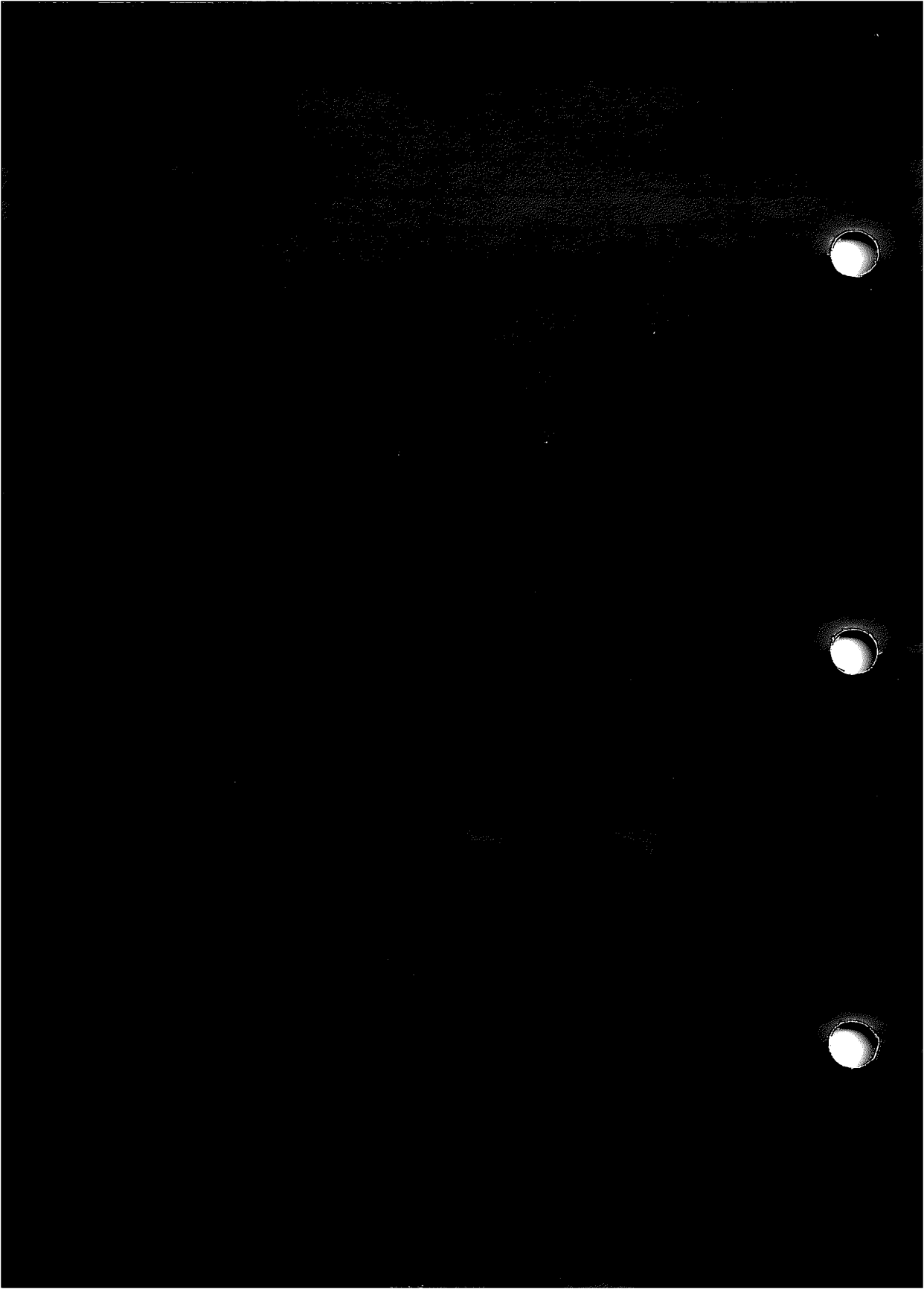
To install the PRO/EDT editor on your system do the following:

1. Log into a system manager's account or privileged user's account.
2. Install the Command Language application if it has not been installed. Refer to the section in this manual on Installing DCL on a Hard Disk System.
3. Select the Command Language application from the main menu or application group you installed it in.
4. Insert the diskette labeled P/OS HARD DISK PRO SERVICE APPLICATIONS into one of the diskette drives.
5. Type the DCL command:

```
@PROSE:[PROEDT]EDT EDT  
INSTAL
```

PRO/EDT is now installed.

Appendix C



Appendix C

Device Names

This Appendix describes how devices are named in P/OS and the proper use of those names. To achieve as much configuration independence as possible, device names are used the same way for stand-alone, P/OS Server, and Workstation systems. A device in P/OS may be accessed using its physical device name in the form:

DDnnn:

where:

DD is two characters.

nnn is the unit number of the device.

In addition, devices may be accessed using logical names. Accessing the devices using logical names is the recommended method.

LOCAL DISKETTES

Names for diskettes local to the Workstation (or to a stand-alone system or a P/OS Server used as a Workstation) are named DZ1 and DZ2. DZ1 refers to the diskette in diskette drive 1. DZ2 refers to drive 2.

Use the diskette's volume label followed by a colon to refer to a diskette by name, regardless of which drive it is in, for example, volumelabel:.

No file/directory protection checks are performed when accessing diskettes. Possession of the diskette is sufficient to guarantee access.

HARD DISKS

A local hard disk (if any) is named DW2:. Should another local hard disk be present, it is named DW3:.

Hard disks—both local and on the P/OS Server—can also be accessed using names in the form:

node\$\$volumelabel:

where:

node	is the DECNET node name of the system (Workstation or P/OS Server) that controls the disk.
volumelabel	is the disk's volume label.

Note that this name remains the same regardless of whether the access is performed from a Workstation or directly on the P/OS Server. However, it will change if the system's node name changes.

Physical device names such as DW002: can be translated to obtain the equivalent node\$\$volumelabel: form.

Access to local hard disks is controlled by the protection [group,member] UIC assigned to the current user's account on the legal system. Access to P/OS Server hard disks is controlled by the protection [group,member] (UIC) assigned to the current user's account on the P/OS Server. The protection [group,member] UIC is assigned when creating an account using the Account Management utility.

The User's Personal Area

To preserve on disk the user and application environments that existed on previous P/OS systems, a personal area with a structure similar to a complete hard disk is defined for each user. For Workstation users, this area is normally located on one of the Server's hard disks so that the user may move among the Workstations freely. Optionally, this area may be defined to reside on a local Workstation disk if one exists.

This personal area is named DW1:, which is a logical name with an equivalence name in the form:

FRnnn:

where:

FR are the letters FR.
 nnn is the unit number of the device.

It may also be referred to by the name:

node\$\$volumelabel\$account:

where:

node is the DECNET node name of the system
 (Workstation or P/OS Server) that controls the disk.
 volumelabel is the associated hard disk.
 account is the user account name associated with a given per-
 sonal area.

The name, node\$\$volumelabel\$account: remains stable unless the associated node name changes. Neither this name nor the FRnnn: form is defined outside the active context of a given user.

FRnnn: may be translated to obtain the node\$\$volumelabel\$account: logical. The special logical name USER\$HOME: may also be translated to obtain this form. Access control is the same as that for the associated hard disk.

SPECIAL DEVICES

To minimize any impact on existing applications, device names that refer to the hard disk have remained the same. The following lists disk device names and their translations that enable access to the user's personal area.

Table C-1
Device Names and Their Translations

<i>Logical Name</i>	<i>Translation</i>
BIGVOLUME:	__DW001:
DW001:	BIGVOLUME:
LDW001:	BIGDISK: (or foreign language equivalent)
BIGDISK:	__DW001:
SYSDISK:	__DW001:

Thus, for example, an access to BIGVOLUME:[USERFILES] is equivalent to an access to USER\$HOME:[USERFILES] or DW1:[USERFILES].

Using LB:

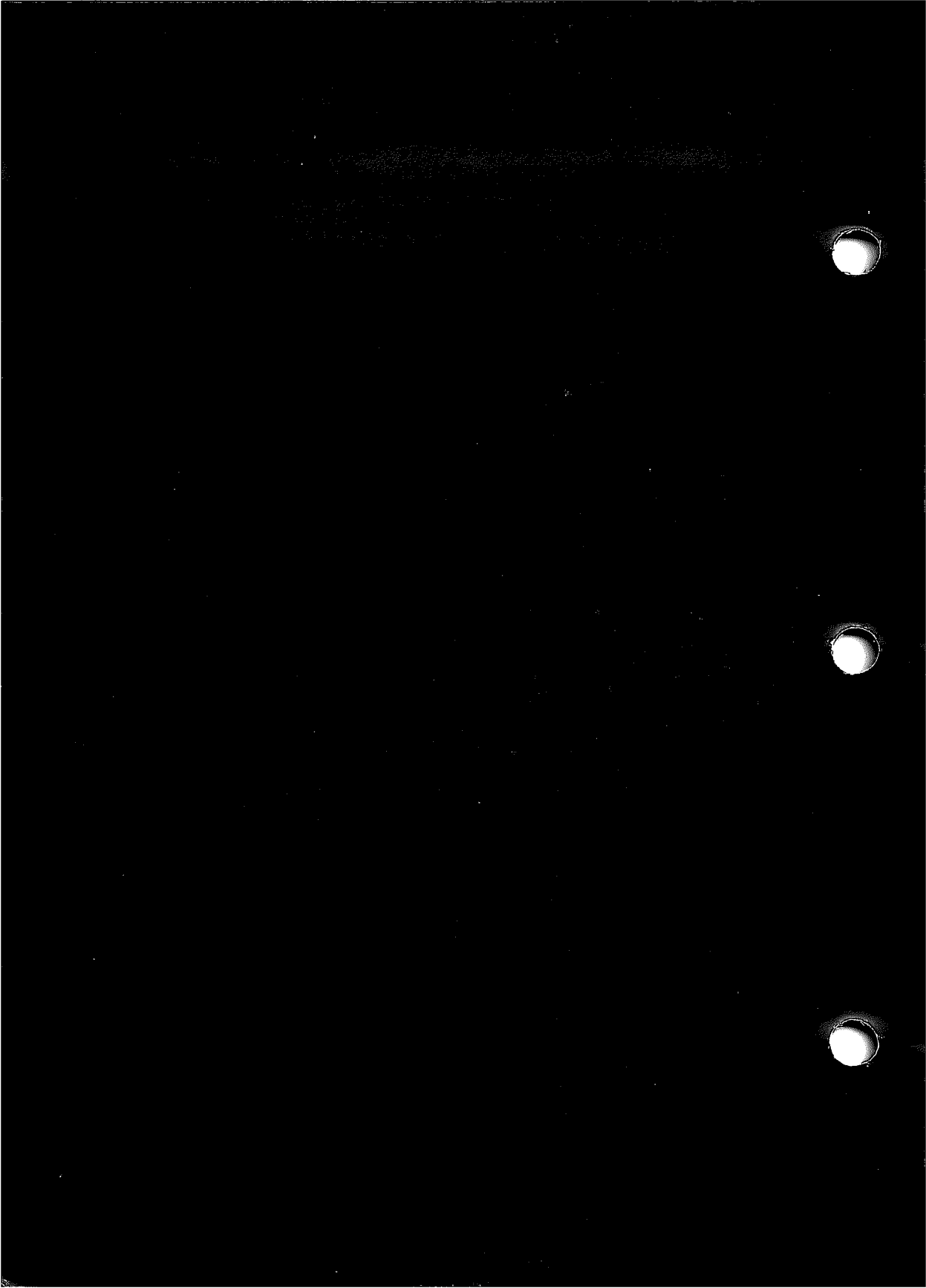
LB: is the traditional P/OS (RSX) library device. In a P/OS Server environment, it is shared by the P/OS Server and all Workstations booted by the P/OS Server. Most system components are located in LB:. Most (often concurrent) access to the LB: device is read-only.

From a Workstation, all access to the LB: device is performed over a special system channel. This channel allows access only to directories and files owned by UIC group 377. Attempts to access files or directories on LB: with different owner-group specifications will fail.

LB1: is an extension of the traditional library device. It provides each individual system with a private library area in addition to the shared library area provided by LB:. Examples of system components on the LB1: device include the account file that controls access to a system, as well as data files that describe hardware options and setup private to that system.

From a Workstation, all access to the LB1: device is performed over the same system channel used for LB:. Thus access is limited to directories and files owned under group 377. Each Workstation system channel has a unique member UIC within group 377. All directories and files on LB1: are usually owned by that [377,member] UIC, with no group or world access permitted. This excludes other Workstations from access to the private library area.

Index



Index

\$, 13
\$-, 17
%, 21
*, 19
-, 17
/, 15
:, 15
< >, 19
=, 16
?, 16

Abbreviating commands, 17
ABORT command, 32
Access control, 24
Account
 in node specification, 25
Alias, 24
/ALLOCATION:N[.] qualifier
 with COPY, 56
APPEND command, 33
Archiving files, 27
Arguments, 15
ARROW keys, 14
ASCII argument
 with APPEND, 35
 with COPY, 58
ASSIGN command, 37
ASSIGN/TASK command, 41
Asterisk (*), use of, 19, 20
At sign (@), use of, 27

/ATTRIBUTES qualifier
 with DIRECTORY, 81
AUTOMATIC argument
 with COPY, 59

BASIC command, 42
BIGDISK:, 18
BIGVOLUME:, 18
BLOCK argument
 with APPEND, 36
 with COPY, 59
Braces, use of, 19
/BRIEF qualifier
 with DIRECTORY, 80

CANCEL command, 43
CANCEL key, 14
Canceling a command, 17
Circumflex (^)
 use in error messages, 171
Cleaning up your files, 27
CLEAR command, 44
.CMD files, 27
COBOL command, 45
Colon (:), use of, 15
Command format, 13, 14
Command name, 14
/COMMAND qualifier
 with RUN, 119, 122
Commands, 31
 ABORT, 32
 APPEND, 33

INDEX

- ASSIGN, 37
- ASSIGN/TASK, 41
- BASIC, 42
- CANCEL, 43
- CLEAR, 44
- COBOL, 45
- CONTINUE, 52
 - continuing to next line, 17
- COPY, 53
- CREATE, 60
- CREATE/DIRECTORY, 63
- DEASSIGN, 65
- DEFINE, 67
- DELETE, 71
- DELETE/DIRECTORY, 74
- DIBOL, 75
- DIRECTORY, 78
- DISMOUNT, 83
- EDIT, 84
- EDIT/EDT, 86
- EDIT/PROSE, 84
- EXIT, 14, 89
- FIX, 90
- FORMAT, 92
- FORTRAN, 93
- HELP, 14, 99
- INITIALIZE, 101
- INSTALL, 102
 - list of, 5
- MAIL, 104
- maximum length, 17
- MOUNT, 105
 - overview of, 5
- PASCAL, 106
- PHONE, 109
- PRINT, 110
- PRINT/REMOTE, 111
- PURGE, 112
- REMOVE, 115
- RENAME, 116
- RUN installed task, 119
- RUN uninstalled task, 122
- SET DAYTIME, 124
- SET DEFAULT, 125
- SET DEVICE, 126
- SET HOST, 128
- SET PRIORITY, 129
- SET PROTECTION, 130, 131
- SET PROTECTION/DEFAULT, 134
- SET PROTECTION/NODEFAULT, 136
- SET TERMINAL, 137
- SET TIME, 124
- SHOW CLOCK_QUEUE, 142
- SHOW COMMON, 143
- SHOW DAYTIME, 145
- SHOW DEFAULT, 146
- SHOW DEVICES, 147
- SHOW LOGICALS, 148
- SHOW PROTECTION, 149
- SHOW TASK/LOGICAL_UNITS, 156
- SHOW TASKS/ACTIVE, 150
- SHOW TASKS/INSTALLED, 154
- SHOW TERMINAL, 157
- SHOW TIME, 145
- SPAWN, 159
- START, 160
- START/UNBLOCK, 161
- STOP/BLOCK, 162
- SUBMIT/REMOTE, 163
 - summary of, 5
- TYPE, 164
- UNLOCK, 167
- /CONCATENATE qualifier
 - with COPY, 57
- /CONTIGUOUS qualifier
 - with COPY, 55
- CONTINUE command, 52
- Continuing a command to next line, 17
- COPY command, 53
- Correcting mistakes in commands, 14
- CREATE command, 60
- CREATE/DIRECTORY command, 63
- CTRL/C, use of, 17
- CTRL/Z, use of, 17
- /DATA_TYPE qualifier
 - with APPEND, 35
 - with COPY, 54
- /DATE qualifier
 - with APPEND, 34
 - with COPY, 55
 - with DELETE, 72
 - with DIRECTORY, 81
 - with PURGE, 113
 - with RENAME, 117
 - with TYPE, 165
 - with UNLOCK, 167
- Dates, how to enter, 124
- DCL, 3
 - command format, 13, 14
 - commands, 31
 - description of, 3
 - displaying help, 14
 - error messages, 13, 171
 - exiting, 14
 - how to correct a mistake, 14
 - how to enter a command, 13
 - how to stop a command, 17
 - indirect commands, 26
 - overview of commands, 5
 - use in cleaning up files, 27

- using PROSE with, 84
- dd-mmm-yy, format for dates, 124
- DEASSIGN command, 65
- DECnet, 3
- Defaults
 - .CMD, 27
 - command file type, 27
 - definition of, 18
 - device/volume, 18
 - directory, 19
 - file type, 19
 - .TSK, 102
 - version number, 19
 - wildcards, 19
 - with DIRECTORY command, 79
 - with INSTALL command, 102
 - with PURGE command, 113
- DEFINE command, 67
- DELETE command, 71
- DELETE/DIRECTORY command, 74
- Deleting parts of commands, 14
- Device, 18
 - initial default, 18
- Devices
 - using, 193
- DIBOL command, 75
- Digital Command Language, 3
- Directory, 19
 - initial default, 19
- DIRECTORY command, 78
- Directory formats
 - foreign files, 25
- Diskette system
 - running DCL, 4
- DISKETTE1:, 18
- DISKETTE2:, 18
- DISMOUNT command, 83
- Displaying helpful information, 14
- Displaying previous commands, 14

- EDIT command, 84
- EDIT/EDT command, 86
- EDIT/PROSE command, 84
- Entering a DCL command, 13
- Equal sign, use of, 16
- Error messages, 13, 171
- /EXCLUDE qualifier
 - with APPEND, 34
 - with COPY, 56
 - with DELETE, 73
 - with DIRECTORY, 82
 - with PURGE, 114
 - with RENAME, 117
 - with SET PROTECTION, 133

- with TYPE, 165
- with UNLOCK, 168
- EXIT command, 14, 89
- EXIT key, 14

- F13 key, 14
- File name, 19
- File specification, 18
 - device, 18
 - directory, 19
 - filename, 19
 - for network operations, 24
 - format, 18
 - initial default directory, 19
 - type, 19
 - version, 19
 - volume, 18, 19
 - wildcards, 19
- File type, 19
- /FINAL qualifier
 - with ASSIGN, 39
 - with DEFINE, 69
- FIX command, 90
- /FIX qualifier
 - with INSTALL, 103
- Foreign files
 - directory formats, 25
- FORMAT command, 92
- Format for commands, 13, 14
- Format for dates, 124
- Format for time, 124
- FORTRAN command, 93
- /FREE qualifier
 - with DIRECTORY, 81
- /FULL qualifier
 - with DIRECTORY, 80

- Hard disk system
 - installing DCL, 4
- HELP command, 14, 99
- HELP key, 14
- hh:mm[:ss], format for time, 124
- Hyphen, use of, 17

- IMAGE argument
 - with APPEND, 35
 - with COPY, 58
- Indirect command files, 26
 - .CMD, 27
- Initial default device, 18
- INITIALIZE command, 101
- INSTALL command, 102

INDEX

- Installation
 - PRO/EDT editor, 189
- Installing DCL on hard disk, 4
- Invoking a file, 26

- /KEEP qualifier
 - with PURGE, 113

- LB:[1,2]EDTSYS.EDT
 - see PRO/EDT use with /COMMAND qualifier
- /LI switch, with RUN command, 119, 122
- /LOG qualifier
 - with APPEND, 35
 - with COPY, 58
 - with DELETE, 72
 - with PRINT/REMOTE, 111
 - with PURGE, 112
 - with RENAME, 118
 - with SUBMIT/REMOTE, 163
 - with TYPE, 166

- /MACY11 qualifier
 - with APPEND, 35
 - with COPY, 58
 - with TYPE, 166
- MAIL command, 104
- Maximum command line, 17
- Messages, 13
- Mistakes, 13
- MOUNT command, 105

- /NEW__VERSION qualifier
 - with COPY, 58
 - with RENAME, 118
- /[NO]BAD__BLOCKS qualifier
 - with INITIALIZE, 101
- /[NO]WRITEBACK qualifier
 - with INSTALL, 103
- /NOCONFIRM qualifier
 - with FORMAT, 92
 - with INITIALIZE, 101
- /NOCONTIGUOUS qualifier
 - with COPY, 55
- Node specification, 24
 - account, 25
 - password, 25
 - userid, 24
- Nodename, 24
- /NOLOG qualifier
 - with PURGE, 112
- /NONEW__VERSION qualifier
 - with COPY, 56
- /NOREMOVE qualifier
 - with INSTALL, 103
- /NOSPAN qualifier
 - with COPY, 58
- /NOWARNINGS qualifier
 - with APPEND, 35
 - with COPY, 56
 - with CREATE/DIRECTORY, 63
 - with DELETE, 73
 - with DIRECTORY, 82
 - with PRINT/REMOTE, 111
 - with PURGE, 114
 - with RENAME, 117
 - with SUBMIT/REMOTE, 163
 - with TYPE, 165
 - with UNLOCK, 167

- /OUTPUT qualifier
 - with DIRECTORY, 81
- /OVERLAY qualifier
 - with COPY, 56
- Overview of commands, 5
- /OWN qualifier
 - with COPY, 57
- /OWNER__UIC:[UIC] qualifier
 - with CREATE/DIRECTORY, 64

- P/OS, 3
- Parameters, 15
- PASCAL command, 106
- Password
 - in node specification, 25
- Percent sign, use of, 21
- PHONE command, 109
- /PRESERVE__DATE qualifier
 - with COPY, 57
- PRINT command, 110
- /PRINT qualifier
 - with APPEND, 36
 - with COPY, 59
- PRINT/REMOTE command, 111
- PRO/DECnet, 3
- PRO/EDT
 - command file
 - see Appendix B
 - directory location, 86
 - use with /COMMAND qualifier, 86
- PRO/EDT editor
 - installing, 189
- Professional Operating System, 3
- Prompts, 16

- \$, 13
- \$-, 17
- ?, 16
 - continuation, 17
 - File(s)?, 16
 - Old File Name?, 16
- PROSE
 - Exiting, 85
- PROSE, running, 84
- PURGE command, 112

- Qualifier, 15
- /QUERY qualifier
 - with DELETE, 72

- /READONLY__COMMON qualifier
 - with INSTALL, 103
- RECORD argument
 - with APPEND, 36
 - with COPY, 59
- /REGION qualifier
 - with REMOVE, 115
- REMOVE command, 115
- RENAME command, 116
- /REPLACE qualifier
 - with COPY, 55
- RETURN key, to continue commands, 17
- RMS-11 files, 18
- RUN installed task command, 119
- RUN uninstalled task command, 122
- Running DCL on diskette system, 4

- /SESSION qualifier
 - with ASSIGN, 39
 - with DEFINE, 69
- SET DAYTIME command, 124
- SET DEFAULT command, 125
- SET DEVICE command, 126
- SET HOST command, 128
- SET PRIORITY command, 129
- SET PROTECTION command, 130, 131
- SET PROTECTION/DEFAULT command, 134
- SET PROTECTION/NODEFAULT command, 136
- SET TERMINAL command, 137
- SET TIME command, 124
- /SHARED qualifier
 - with COPY, 57
 - with TYPE, 165
- SHOW CLOCK__QUEUE command, 142
- SHOW COMMON command, 143
- SHOW DAYTIME command, 145
- SHOW DEFAULT command, 146
- SHOW DEVICES command, 147
- SHOW LOGICALS command, 148
- SHOW PROTECTION command, 149
- SHOW TASK/LOGICAL__UNITS command, 156
- SHOW TASKS/ACTIVE command, 150
- SHOW TASKS/INSTALLED command, 154
- SHOW TERMINAL command, 157
- SHOW TIME command, 145
- /SINCE qualifier
 - with APPEND, 34
 - with COPY, 55
 - with DELETE, 72
 - with DIRECTORY, 81, 82
 - with PURGE, 113
 - with RENAME, 117
 - with SET PROTECTION, 132
 - with TYPE, 165
 - with UNLOCK, 167
- Single line editor, 14
- Slash (/), use of, 15
- Space, use of in commands, 15
- SPAWN command, 159
- START command, 160
- START/UNBLOCK command, 161
- STOP/BLOCK command, 162
- Stopping a command, 17
- /SUBMIT qualifier
 - with COPY, 59
- SUBMIT/REMOTE command, 163
- Summary of commands, 5
- /SUMMARY qualifier
 - with DIRECTORY, 80
- Syntax for commands, 13
- /SYSTEM qualifier
 - with ASSIGN, 39
 - with DEFINE, 69

- /TASK__NAME qualifier
 - with INSTALL, 103
 - with RUN, 123
- Terminating command execution, 17
- /THROUGH qualifier
 - with APPEND, 34
 - with COPY, 56
 - with DELETE, 72
 - with DIRECTORY, 81
 - with PURGE, 113
 - with RENAME, 117
 - with SET PROTECTION, 133
 - with TYPE, 165
 - with UNLOCK, 168
- Time, how to enter, 124

INDEX

- /TODAY qualifier
 - with APPEND, 35
 - with COPY, 56
 - with DELETE, 72
 - with DIRECTORY, 82
 - with PURGE, 113
 - with RENAME, 117
 - with SET PROTECTION, 133
 - with TYPE, 165
 - with UNLOCK, 168
- /TRANSFER__MODE qualifier
 - with APPEND, 36
 - with COPY, 59
- /TRANSLATION__ATTRIBUTES qualifier
 - with ASSIGN, 39
 - with DEFINE, 69
- .TSK, 102
- Type
 - default for INSTALL command, 102
 - defaults, 19
 - file, 19
- TYPE command, 164
- UIC, 181
- UNLOCK command, 167
- /USER qualifier
 - with ASSIGN, 39
 - with DEFINE, 69
- USERDISK, 18
- Userid
 - in node specification, 24
- Version number, 19
- Volume, 18
- /WIDTH qualifier
 - with DIRECTORY, 81
- Wildcards, 19
 - with /EXCLUDE qualifier, 73
 - with COPY command, 54
 - with DELETE command, 71
 - with DIRECTORY command, 79

READER'S COMMENTS

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- First-time computer user
- Experienced computer user
- Application package user
- Programmer
- Other (please specify) _____

Name _____

Date _____

Organization _____

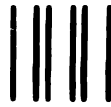
Street _____

City _____

State _____

Zip Code
or Country _____

digital

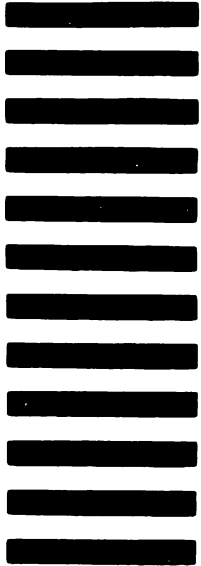


No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 33 MAYNARD MASS

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN: Professional Workstations Publications
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET ML021-2/T76
MAYNARD, MA 01754-2571



Do Not Tear - Fold Here

Cut Along Dotted Line



Printed in Holland
AA-DY40A-TH