

RSX-11M
Command Language Manual

Order No. AA-L672C-TC

digital
software

RSX-11M
Command Language Manual

Order No. AA-L672C-TC

RSX-11M Version 4.2

First Printing, October 1979
Revised, November 1981
Revised, April 1983
Revised, July 1985

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 be used or copied only 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 Equipment Corporation or its affiliated companies.

Copyright © 1979, 1981, 1983, 1985 by Digital Equipment Corporation
All Rights Reserved.

Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	PDT
DEC/CMS	EduSystem	RSTS
DEC/MMS	IAS	RSX
DECnet	MASSBUS	UNIBUS
DECsystem-10	MicroPDP-11	VAX
DECSYSTEM-20	Micro/RSTS	VMS
DECUS	Micro/RSX	VT
DECwriter	PDP	

digital

ZK2868

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710
In New Hampshire, Alaska, and Hawaii call 603-884-6660
In Canada call 613-234-7726 (Ottawa-Hull)
800-267-6215 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575)

DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation
PSG Business Manager
c/o Digital's local subsidiary or
approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

CONTENTS

	Page
PREFACE	vii
SUMMARY OF TECHNICAL CHANGES	xi
CHAPTER 1	HOW TO USE DCL
1.1	RSX-11M AND DCL: BASIC CONCEPTS 1-1
1.2	THE DCL COMMAND LINE 1-2
1.2.1	Prompting 1-3
1.2.2	Qualifiers 1-3
1.2.3	HELP 1-4
1.2.4	MCR and DCL Commands 1-6
1.2.5	Abbreviations 1-7
1.2.6	Numbers and Dates 1-8
1.2.7	Multiple Parameters 1-8
1.2.8	Underscore Character 1-8
1.2.9	Colon and Equal Sign 1-9
1.2.10	Quoting Strings 1-9
1.2.11	ANSI File Names 1-9
1.2.12	DECnet and DCL 1-10
1.2.12.1	SET HOST 1-10
1.2.12.2	SHOW HOST 1-11
1.2.13	Command Line Continuation 1-12
1.2.14	Comments in Command Lines 1-12
1.2.15	SET DEBUG 1-13
1.2.16	Errors 1-14
1.3	FUNCTIONAL GROUPING OF DCL COMMANDS 1-15
1.3.1	The SET and SHOW Commands 1-15
1.3.2	Terminal Operations 1-16
1.3.3	Handling Files 1-16
1.3.4	Handling Devices and Volumes 1-16
1.3.5	Program Development 1-16
1.3.6	Running Tasks 1-17
1.3.7	System Control 1-17
1.4	USING THE QUEUE MANAGER: PRINT JOBS 1-17
1.5	INDIRECT COMMAND FILES IN DCL 1-17
1.6	DCL AND RSX-11M UTILITIES AND SYSTEM TASKS 1-18
1.7	RSX-11M DCL AND VAX/VMS DCL 1-19
CHAPTER 2	THE SET AND SHOW COMMANDS
2.1	SET AND SHOW TIME 2-1
2.2	SHOW MEMORY 2-2
2.3	SET AND SHOW TERMINAL 2-2
2.4	SHOW DEVICES 2-3
2.5	SHOW QUEUE 2-4
2.5.1	SHOW PROCESSOR 2-4
2.6	SHOW ASSIGNMENTS 2-5
2.7	SET AND SHOW DEFAULTS 2-5
2.8	SHOW TASKS 2-5
2.9	SHOW USERS 2-6
2.10	SET PROTECTION 2-6
2.11	SET AND SHOW SYSTEM/DIRECTORY 2-8
2.12	SET DEBUG 2-8
CHAPTER 3	TERMINAL OPERATIONS
3.1	LOGGING IN AND LOGGING OUT 3-2
3.1.1	LOGIN 3-4

CONTENTS

3.1.2	LOGOUT	3-8
3.1.3	ACNT, the Account File Maintenance Program . . .	3-9
3.2	THE KEYBOARD	3-9
3.2.1	RETURN and Command Line Length	3-10
3.2.2	Line Terminators	3-10
3.2.3	DELETE Versus BACK SPACE	3-11
3.2.4	CAPS LOCK and SHIFT	3-12
3.2.5	REPEAT	3-12
3.2.6	SCROLL and NO SCROLL	3-12
3.2.7	Other Keys	3-12
3.3	THE KEYPAD	3-15
3.4	THE CTRL KEY AND CONTROL CHARACTERS	3-16
3.4.1	CTRL/C	3-17
3.4.2	CTRL/S and CTRL/Q	3-17
3.4.3	CTRL/O	3-18
3.4.4	CTRL/Z	3-18
3.4.5	CTRL/U	3-19
3.4.6	CTRL/R	3-19
3.4.7	CTRL/X	3-20
3.4.8	CTRL/L	3-20
3.4.9	Other Control Characters	3-20
3.5	HELP	3-21
3.5.1	Help File Format: Writing a Help File	3-24
3.6	BROADCAST	3-28
3.7	REQUEST	3-32
3.8	SET AND SHOW TERMINAL	3-33
3.8.1	SET TERMINAL	3-34
3.8.2	SHOW TERMINAL	3-47

CHAPTER 4 HANDLING FILES

4.1	FILES ON RSX-11M SYSTEMS	4-1
4.1.1	File Ownership and Location	4-2
4.1.2	File Specification	4-2
4.1.3	Defaults in File Specifications	4-4
4.1.4	Wildcards in File Specifications	4-6
4.1.5	Date-Related Qualifiers	4-8
4.1.6	File Protection and Volume Protection	4-9
4.2	CREATING DIRECTORIES AND SEQUENTIAL FILES	4-10
4.2.1	CREATE	4-11
4.2.2	CREATE/DIRECTORY	4-13
4.2.3	EDIT	4-15
4.2.3.1	EDIT[/EDT]	4-16
4.2.3.2	EDIT/SLP	4-20
4.2.3.3	EDIT/EDI and EDIT/editor	4-23
4.3	MAINTAINING A DIRECTORY	4-25
4.3.1	DIRECTORY	4-26
4.3.2	DELETE	4-35
4.3.3	PURGE	4-39
4.3.4	COPY	4-42
4.3.5	RENAME	4-47
4.3.6	CONVERT	4-50
4.4	OTHER FILE-RELATED COMMANDS	4-56
4.4.1	TYPE	4-57
4.4.2	PRINT	4-60
4.4.3	APPEND	4-70
4.4.4	DIFFERENCES	4-72
4.4.5	UNLOCK	4-78
4.5	SET AND SHOW	4-79
4.5.1	SET DEFAULT	4-80
4.5.2	SHOW DEFAULT	4-82
4.5.3	SET PROTECTION	4-83
4.5.4	SET FILE	4-87

CONTENTS

CHAPTER 5	HANDLING DEVICES AND VOLUMES	
5.1	DEVICES ON RSX-11M SYSTEMS	5-1
5.1.1	Volumes and Devices	5-4
5.1.2	Physical Devices, Pseudo Devices, LUNs, and Logical Names	5-5
5.1.2.1	Pseudo Devices	5-5
5.1.2.2	LUNs	5-7
5.1.2.3	Logical Device Names	5-8
5.1.3	Public, Shared, Private and Unowned Devices and Mounted Volumes	5-8
5.1.4	How to Prepare a Scratch Disk for Use	5-9
5.1.5	Preparing a Scratch ANSI Magnetic Tape for Use	5-12
5.1.6	Summary of Device and Volume Use Commands	5-12
5.2	ANALYZE/MEDIA	5-14
5.3	ASSIGN	5-21
5.3.1	ASSIGN/REDIRECT	5-24
5.3.2	ASSIGN/TASK	5-27
5.4	DEASSIGN	5-29
5.5	ALLOCATE	5-31
5.6	DEALLOCATE	5-34
5.7	MOUNT	5-36
5.7.1	Mounting Magnetic Tapes and Tape Sets	5-48
5.8	DISMOUNT	5-53
5.9	INITIALIZE	5-57
5.9.1	INITIALIZE/UPDATE	5-70
5.10	BACKUP	5-73
5.11	SET AND SHOW	5-85
5.11.1	SHOW ASSIGNMENTS	5-86
5.11.2	SET DEVICE	5-88
5.11.3	SHOW DEVICES	5-91
CHAPTER 6	PROGRAM DEVELOPMENT	
6.1	INTRODUCTION	6-1
6.2	LINKING THE TASK	6-2
6.2.1	Introduction to the LINK Command	6-2
6.2.2	LINK	6-7
6.3	LIBRARY	6-28
6.3.1	LIBRARY/COMPRESS	6-31
6.3.2	LIBRARY/CREATE	6-32
6.3.3	LIBRARY/DELETE	6-35
6.3.4	LIBRARY/EXTRACT	6-36
6.3.5	LIBRARY/INSERT	6-37
6.3.6	LIBRARY/LIST	6-39
6.3.7	LIBRARY/REMOVE	6-43
6.3.8	LIBRARY/REPLACE	6-44
6.3.9	Error Messages from LIBRARY	6-45
6.4	SET AND SHOW	6-51
6.4.1	SET GROUPFLAGS	6-52
6.4.2	SHOW GROUPFLAGS	6-53
CHAPTER 7	RUNNING TASKS	
7.1	TASK INSTALLATION AND EXECUTION	7-1
7.1.1	Task Naming	7-2
7.2	INTRODUCTION TO THE RUN COMMAND	7-5
7.2.1	Running Tasks Contained in Task Image Files	7-7
7.2.2	Running and Scheduling Installed Tasks	7-11
7.3	ABORT	7-16
7.4	CONTINUE	7-21
7.5	CANCEL	7-23
7.6	STOP/BLOCK	7-24
7.7	START	7-26

CONTENTS

7.7.1	START/UNBLOCK	7-27
7.8	INSTALL	7-28
7.9	REMOVE	7-37
7.10	FIX	7-38
7.11	UNFIX	7-40
7.12	SET AND SHOW	7-40
7.12.1	SET PARTITION	7-41
7.12.2	SHOW PARTITIONS	7-46
7.12.3	SHOW TASKS	7-49
7.12.3.1	SHOW TASKS/ACTIVE	7-49
7.12.3.2	SHOW TASKS/INSTALLED	7-54
7.12.3.3	SHOW TASKS/INSTALLED/DEVICE	7-56
7.12.3.4	SHOW TASKS/DYNAMIC	7-57
7.12.3.4.1	SHOW TASK:taskname/DYNAMIC	7-57
7.12.3.4.2	SHOW TASKS/ACTIVE/DYNAMIC	7-59
7.12.3.5	SHOW TASK/LOGICAL_UNITS	7-61
7.12.4	SET PRIORITY	7-63
7.12.5	SHOW /CLOCK_QUEUE	7-65

CHAPTER 8 SYSTEM CONTROL

8.1	SET AND SHOW	8-1
8.1.1	SET DAYTIME	8-1
8.1.2	SHOW [DAY]TIME	8-3
8.1.3	SET SYSTEM	8-3
8.1.4	SHOW SYSTEM	8-6
8.1.5	SYSTEM DIRECTORIES	8-7
8.1.6	SHOW USERS	8-8
8.1.7	SHOW MEMORY	8-9

CHAPTER 9 QUICK REFERENCE

APPENDIX A COMMON ERROR MESSAGES

APPENDIX B FUNCTIONS INITIATED BY DCL COMMANDS

INDEX

FIGURES

3-1	Terminal Keyboard and Keypad	3-1
4-1	File Organization on a Volume	4-1
4-2	Job Flag Page	4-62
4-3	File Flag Page	4-63

TABLES

3-1	Keyboard and Keypad Characters	3-13
3-2	Display of Terminal Attributes from SHOW TERMINAL	3-49
4-1	File Types	4-5
5-1	Devices on RSX-11M Systems	5-2
5-2	Pseudo Devices on RSX-11M	5-6
6-3	Task Builder Options	6-16
7-1	Base and Size Specifications for SET PARTITION	7-42
7-2	Task Status Flags	7-51

PREFACE

MANUAL OBJECTIVES

The RSX-11M Command Language Manual provides information needed to use a terminal to do work on an RSX-11M system. Important system concepts are introduced and explained in the context of the DIGITAL Command Language (DCL). DCL is a user-oriented command language with English words and prompting for required command elements.

INTENDED AUDIENCE

This manual is intended for any user of an RSX-11M system. The manual is a reference manual with many tutorial elements, but new users should read the Introduction to RSX-11M first.

A user is anyone who interfaces with the system to perform some task.

There are two kinds of users: privileged and nonprivileged. Privileged users have access to functions that control and modify system operation as well as functions that permit program development and system maintenance. Nonprivileged users have access to functions required for program development and local maintenance only.

STRUCTURE OF THIS MANUAL

The manual is organized around the major operating system functions. Following the introduction, each major function has a chapter to itself. All commands relating to the function are described in the chapter. The last chapter is an annotated alphabetical listing of all DCL commands.

Chapter 1 introduces the basic concepts of DCL, explains how DCL is implemented on RSX-11M, and includes many hints on using DCL. The chapter also includes summaries of the remainder of the manual, as well as general information about DCL, the operating system, and various utilities and system tasks.

Chapter 2 lists the most common forms of SET and SHOW commands.

Chapter 3 covers terminal operations, including logging in and logging out, keyboard terminology and functions, setting and displaying terminal attributes, broadcasting messages, and using the system HELP commands.

Chapter 4 introduces file-handling operations on RSX-11M, including commands to create and delete files, list directories, and rename, copy, type, and print files. Wildcards and defaults are explained.

PREFACE

Chapter 5 covers peripheral devices and introduces the system I/O terminology. The chapter explains the relationships between hardware devices and the software file system, and describes how to prepare scratch disks and magnetic tapes for use on the system. Commands affecting devices and software volumes are described.

Chapter 6 introduces the program-development facilities of the system and the DCL commands used with these facilities.

Chapter 7 explains how tasks run in the system and how they are named, installed, fixed in memory, and controlled while running. In addition, the chapter describes commands to abort tasks, to place tasks in the clock queue, and to display information about tasks in the system.

Chapter 8 covers the DCL system-control commands.

Chapter 9 is the alphabetical listing of all DCL commands and their formats, including comments to remind you of specific points about many commands and command elements. Each entry in Chapter 9 includes a cross-reference to the full command description in an earlier chapter.

Appendix A explains error messages common to several DCL commands. Other command-specific error messages are explained in the full command descriptions.

Appendix B lists the relationships between DCL and the rest of the operating system.

ASSOCIATED MANUALS

Other documents related to RSX-11M are described in the RSX-11M/RSX-11S Information Directory and Index.

If you are entirely new to computers or to DCL, you should read the Introduction to RSX-11M and follow the instructions there.

Although this manual is meant to be self-contained, you will find important information in the RSX-11M/M-PLUS MCR Operations Manual and the RSX-11M/M-PLUS Utilities Manual.

Programmers should see the RSX-11M/M-PLUS Guide to Program Development.

CONVENTIONS USED IN THIS MANUAL

A number of conventions are used in the command descriptions in this manual:

Convention

Meaning

CTRL/a

A symbol that indicates the CTRL key; it must be held down while another key is pressed. For example, CTRL/Z means hold down the CTRL key while pressing Z. In examples, this control key sequence is shown as ^a; for example, ^O indicates the result of CTRL/O because that is how the system echoes most control key combinations.

xxx

A 1- to 3-character key symbol. For example, RET indicates the RETURN key, LF indicates the LINE FEED key, and ESC indicates the ESCAPE key.

PREFACE

- . The vertical ellipsis shows where elements of command input have been omitted because they are irrelevant to the point being discussed.
- .
[/*qualifier*] Any command field enclosed in brackets is optional. If the brackets include syntactical elements, such as dots (.) or slashes (/), those elements are required for the field. If the field appears in lowercase, you are to substitute a legal command element if you include the field.
- [*g,m*] This signifies either a User Identification Code (UIC) or a directory. The *g* is a group number and the *m* is a member number. Where a UIC or directory is required, only one set of brackets is shown, as [*g,m*]. Where the UIC or directory is optional, two sets of brackets are shown, as [[*g,m*]]. The UIC identifies a user and is used mainly for controlling access to files and privileged system functions.
- UPPERCASE Any command field in uppercase indicates the legal form of the command. If you type it in that form, it will work as described. Most DCL commands have abbreviations.
- lowercase Any command field in lowercase is to be substituted for. Usually the lowercase word identifies the kind of substitution expected, such as *filespec*, which indicates that you should fill in the file specification.
- /*qualifier* Any command element preceded by a slash (/) is a DCL qualifier. Command qualifiers alter the action of a command they are attached to. Parameter qualifiers modify the action of the command as it affects that parameter.
- parameter Required command fields are generally called parameters. The most common parameters are file specifications. Parameters are preceded by blanks or DCL prompts.
- :*argument* Some parameters and qualifiers can be altered by the inclusion of arguments preceded by a colon. An argument can be either numerical (COPIES:3) or alphabetical (NAME:KAT). The equals sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.
- filespec* A full file specification includes device, directory, file name, file type, and version number, as in this example:

```
DB1:[303,5]HIPPIY.HOP;2
```

Full file specifications are rarely needed. If you do not give a version number, the highest numbered version will be used. If you do not give a directory, the default directory will be used. Some system functions default to particular file types. See Chapter 4 for more information on file specifications. See also the individual command descriptions.

PREFACE

Some commands accept a file specification with a DECnet node name. See Chapter 1 for more information.

red ink

All user input in examples is printed in red ink to distinguish it from system output. That is, what you type is shown in red.

{A B}

A number of options between braces and separated by vertical bars means that you are to choose one of the options listed.

See Chapter 1 for more information on DCL conventions.

SUMMARY OF TECHNICAL CHANGES

The RSX-11M Command Language Manual has been revised to reflect changes made to the software for Version 4.2. Not all of the alterations made to the software for this version of RSX-11M resulted in changes to this manual; see the RSX-11M Release Notes for a comprehensive list of all software changes.

The following is a list of DCL commands and qualifiers new to this version of RSX-11M:

ANALYZE/MEDIA[/qualifier[s]] ddn:
 /ALLOCATE
 /BAD_BLOCKS
 /BAD_BLOCKS/EXERCISE:(n,m)
 /BAD_BLOCKS/NOEXERCISE
 /[NO]EXERCISE[:(n,m)]
 /OVERRIDE
 /RETRY
 /SHOW

New command. The ANALYZE/MEDIA command determines if bad blocks exist on a disk volume and records their locations for use by the backup and restore utilities and the INITIALIZE command.

APPEND[/qualifier[s]]
 /NOWARNINGS
 /REWIND
 /SHARED

New qualifiers.

BACKUP[/qualifier[s]]
 /CREATED:arg
 AFTER:(dd-mmm-yy hh:mm)
 BEFORE:(dd-mmm-yy hh:mm)
 /APPEND

New qualifiers.

COPY[/qualifier[s]]
 /ALLOCATION:n
 /NOWARNINGS
 /OVERLAY
 /PRESERVE_DATE
 /REWIND
 /SHARED

New qualifiers.

SUMMARY OF TECHNICAL CHANGES

CREATE/DIRECTORY[/qualifier]
/NOWARNINGS
/OWNER_UIC

New qualifiers.

DELETE[/qualifier[s]]
/[NO]CONFIRM
/NOWARNINGS

New qualifiers.

DIRECTORY[/qualifier[s]]
/NOWARNINGS
/REWIND

New qualifiers.

DISMOUNT[/qualifier[s]]
/SAVE
/TERMINAL
/[NO]UNLOAD

New qualifiers.

INITIALIZE[/qualifier[s]]
/BAD_BLOCKS:arg
AUTOMATIC
(AUTOMATIC,MANUAL)
MANUAL
NOAUTOMATIC
OVERRIDE
(OVERRIDE,MANUAL)
/DENSITY:arg
HIGH
LOW
/LABEL:VOLUME_ACCESSIBILITY:"c"

New qualifiers.

INITIALIZE/UPDATE[/qualifier[s]]
/DENSITY:arg
HIGH
LOW

New arguments.

INSTALL[/qualifier[s]]
/[NO]INTERPRETER

New qualifier.

LINK[/qualifier[s]]
/[NO]PRINT

New qualifier.

SUMMARY OF TECHNICAL CHANGES

```
MOUNT[/qualifier[s]]
      /PROCESSOR:arg
      UNIQUE
```

Changed argument to existing qualifier.

```
PRINT[/qualifier[s]]
      /AFTER:TOMORROW
      /NOWARNINGS
```

New qualifiers.

```
PURGE[/qualifier[s]]
      /NOWARNINGS
```

New qualifier.

```
RENAME[/qualifier[s]]
      /NEW_VERSION
      /NOWARNINGS
```

New qualifiers.

```
RUN[/qualifier[s]]
      /[NO] IO_PAGE
      /[NO] SLAVE
```

New qualifiers.

```
SET DEVICE:ddnn:/qualifier[s]
      /[NO]CHECKPOINT_FILE[:n]
```

New qualifier.

```
SET FILE[/qualifier[s]] filespec[s]
      /END_OF_FILE:(BLOCK:n, BYTE:n)
      /ENTER:synonym_filespec
      /NOWARNINGS
      /REMOVE
      /REWIND
      /TRUNCATE
```

New command. SET FILE establishes certain file attributes.

SET HOST nodename

New command. SET HOST connects your terminal to a remote system. You issue this command after you have logged in to your current system. Both your current and remote system must run DECnet software.

```
SET SYSTEM/qualifier
      /NETWORK_UIC:[g,m]
      /POOL/LIMITS
```

New qualifiers.

SUMMARY OF TECHNICAL CHANGES

```
SET TERMINAL[:ttnn:]/qualifier[s]
    /INQUIRE
    /[NO]ADVANCED_VIDEO
    /[NO]ANSI_CRT
    /[NO]BLOCK_MODE
    /DEC_CRT
    /DTC01
    /EDIT_MODE
    /HOSTSYNC
    /LA50
    /LA210
    /LN03
    /LQP02
    /LQP03
    /[NO]PARITY[:type]
        ODD
        EVEN
    /[NO]PASTHRU
    /PRINTER_PORT
    /PRO_SERIES
    /[NO]REGIS
    /[NO]SOFT_CHARACTERS
    /[NO]TTSYNC
    /VT200_SERIES
```

New qualifiers.

SHOW HOST

New command. SHOW HOST displays the name of the processor to which your terminal currently is connected. It also shows you the name and version number of the operating system running on the processor.

```
SHOW SYSTEM[/qualifier]
    /NETWORK_UIC
    /POOL/LIMITS
```

New qualifiers.

```
SHOW TASKS[:taskname]/qualifier[s]
    /DEVICE:ddnn:
```

New qualifier.

```
SHOW TERMINAL[:ttnn:][/qualifier]
    /[NO]ADVANCED_VIDEO
    /[NO]ANSI_CRT
    /[NO]BLOCK_MODE
    /[NO]DEC_CRT
    /DTC01
    /[NO]EDIT_MODE
    /HOSTSYNC
    /[NO]LA50
    /[NO]LA210
    /LN03
    /LQP02
    /LQP03
    /[NO]PARITY
    /[NO]PASTHRU
```

SUMMARY OF TECHNICAL CHANGES

/PRINTER_PORT
/PRO_SERIES
/[NO]REGIS
/[NO]TTSYNC
/[NO]VT200_SERIES

New qualifiers.

TYPE[/qualifier[s]
/NOWARNINGS
/SHARED

New qualifiers.

UNLOCK[/qualifier[s]]
/NOWARNINGS

New qualifier.

The following is a list of new devices supported by RSX-11M Version 4.2:

DHU11	RQDX2
DEUNA	RQDX3
KDA50	RUX50
RD52	TK25
RD53	TK50

The following is a list of new terminal types supported by RSX-11M Version 4.2:

DTC01	LQP02
LA50	LQP03
LA210	PRO-series terminals
LN03	VT200-series terminals

CHAPTER 1

HOW TO USE DCL

DCL is the DIGITAL Command Language. DCL provides RSX-11M users with an extensive set of commands for interactive program development, device and file manipulation, and interactive program execution and control.

If you are new to computers, or if you wish a quick review of DCL and the operating system, you should read the Introduction to RSX-11M. This manual includes a guided tour of the operating system, including an interactive terminal session that introduces all of the most commonly used DCL commands.

DCL commands are full words that describe the action to be taken, not abbreviations or mnemonics. Thus, if you wish to set a terminal to lowercase, you type SET TERMINAL/LOWERCASE. Because DCL commands and command elements are full words, they are self-documenting.

You are not required to use the full form of DCL commands at the terminal, however. Usually, you need type only the command elements required to form a unique command. For instance, SE TERM/LOW will also set a terminal to lowercase, but SE TERM/LO will not work because there is also a command SET TERMINAL/LOCAL. See Section 1.2 for more information on how DCL commands are parsed.

1.1 RSX-11M AND DCL: BASIC CONCEPTS

RSX-11M systems may have one or more command line interpreters (CLIs). All systems include MCR (the Monitor Console Routine). Many systems also include DCL, and some systems include other CLIs. Both MCR and DCL include commands to invoke most system tasks and utilities and to set and display certain system characteristics. In general, MCR commands name tasks, such as PIP, a utility used to manipulate files (for example, to copy or type them), while DCL commands specify actions directly, as in the COPY command or the TYPE command.

MCR is the fundamental command line interpreter (CLI) for the RSX-11M operating systems. MCR also provides commands to set and display certain system characteristics. MCR provides the most direct interface with the RSX-11M operating system.

In general, MCR commands must be entered in exact syntax. MCR commands follow no set syntax rules, however. Most MCR commands are abbreviations or mnemonics.

DCL is an optional user-oriented CLI included in most systems with many users. Commands in DCL are English-like words and follow well-defined syntax rules. Full commands are self-documenting. DCL is designed for consistency and ease of use.

HOW TO USE DCL

DCL is the command language used on many of DIGITAL operating systems. In particular, RSX-11M DCL is designed for compatibility with future releases of VAX/VMS DCL. See Section 1.7 for more information on DCL compatibility. DCL on RSX-11M systems is a CLI task that translates DCL commands into MCR commands for execution by the system. The DCL command SET DEBUG command displays the MCR translation for any DCL command on your terminal. See Section 1.2.15 for more information on the SET DEBUG command.

Depending on the kind of use you make of your system and the nature of your system, you may find it more convenient to use one CLI or the other, or both. All nonprivileged system functions are available directly from DCL, but some privileged functions are not. All program-development facilities and all common utility functions are available from DCL.

Because RSX-11M is designed to be tailored to the needs of each installation, not every feature of DCL described in this manual is available on every system. Some DCL commands depend on layered products that may not be available at your installation. Many features are system-generation options that may not have been selected at the time your system was generated. You should see your system manager or another knowledgeable system user if some feature described in this manual does not appear to be available at your installation.

1.2 THE DCL COMMAND LINE

A command consists of a command name or verb describing the action the system is to take. Most commands also include one or more parameters and qualifiers to further define the action of the command. Qualifiers are preceded by a slash (/) and parameters are preceded by a space. Both qualifiers and parameters can take arguments. Arguments are preceded by a colon (:).

Commands are passed to the operating system with a terminator, either RETURN or ESC. (The ESC key may be labeled SEL or ALT MODE. Its action differs from that of the RETURN key in that the cursor does not return to the left margin and no monitor prompt is returned after execution is completed.)

Unless the action of the RETURN greatly differs from what is expected, the examples and command formats in this manual do not include an indication that each line in the command is terminated by a carriage return. (See Section 1.2.13, for a discussion of command lines too long to fit on one line of your terminal.)

Some commands require parameters or arguments as part of the command line. If you fail to supply a required command element, DCL prompts you with one or two words indicating the general nature of the required element. If you do not understand the prompt, type a question mark (?) for help. (In some cases, an omission causes an error rather than a prompt.)

Section 1.2.1 discusses prompting in detail.

This section introduces the rules governing the use of DCL. The examples in this section are intended to illustrate these rules, not to illustrate the full capabilities of the command. For more detail, see the individual command descriptions.

1.2.1 Prompting

The prompts teach you the form of a command by requesting that you supply required command elements.

For example, the RENAME command works as follows:

```
$ RENAME (RET)
From? BROWNS.STL (RET)
To? ORIOLES.BLT (RET)
```

The one-line format for RENAME is as follows:

```
$ RENAME BROWNS.STL ORIOLES.BLT (RET)
```

The formats can be mixed. DCL prompts for whatever you leave out. For example:

```
$ RENAME BROWNS.STL (RET)
To? ORIOLES.BLT (RET)
```

There are no defaults for prompts. You must supply a response to any prompt. If you do not wish to continue with the command, press CTRL/Z.

1.2.2 Qualifiers

Qualifiers modify the action of the command. Qualifiers always start with a slash (/) and are generally optional.

Qualifiers are either command qualifiers or parameter qualifiers. Most qualifiers are command qualifiers. In this manual, command qualifiers are always shown as modifying the command verb, as in this example:

```
$ TYPE/TODAY *.HLP (RET)
```

However, most command qualifiers can appear anywhere in the command line. Another name for these qualifiers is floating qualifiers. The following examples illustrate how command qualifiers can float:

```
$ TYPE *.HLP/TODAY (RET)
```

or

```
$ TYPE (RET)
File(s)? *.HLP/TODAY (RET)
```

or

```
$ TYPE (RET)
File(s)? /TODAY (RET)
File(s)? *.HLP (RET)
```

You can mix formats and get exactly the same results, as in the following example:

```
$ TYPE/TODAY (RET)
File(s)? *.HLP/EXCLUDE:HELPF.HLP;* (RET)
```

or

HOW TO USE DCL

```
$ TYPE/TODAY (RET)
File(s)? *.HLP/ (RET)
Qualifier? EXCLUDE:HELPF.HLP;* (RET)
```

or

```
$ TYPE/TODAY (RET)
File(s)? / (RET)
Qualifier? EXCLUDE:HELPF.HLP;* (RET)
File(s)? *.HLP (RET)
```

Note that you are prompted for a qualifier when a slash with no qualifier attached appears on the command line. When you supply the qualifier, do not type the slash again.

Almost all command qualifiers can float in this way. In a few cases, the command qualifier must appear directly after the command it modifies. Whenever two similar commands are described separately, such as ASSIGN and ASSIGN/REDIRECT or CREATE and CREATE/DIRECTORY, the distinguishing qualifier cannot float. Other qualifiers to such commands can still float, however, as in this example:

```
$ CREATE/DIRECTORY (RET)
Device, UFD? [303,5]/ALLOCATION:5 (RET)
```

which is the same as the following example:

```
$ CREATE/DIRECTORY/ALLOCATION:5 [303,5] (RET)
```

Parameter qualifiers, sometimes called file specification qualifiers, cannot float. Usually a parameter qualifier must be attached to a file specification, because most DCL parameters are file specifications.

Many qualifiers can be negated by prefixing NO or - (minus) to the qualifier name. Thus, the following command:

```
$ MACRO/OBJECT SIMPLE.MAC (RET)
```

directs the MACRO-11 Assembler to make an object file, while the following command:

```
$ MACRO/NOOBJECT SIMPLE.MAC (RET)
```

or

```
$ MACRO/-OBJECT SIMPLE.MAC (RET)
```

directs the MACRO-11 Assembler to omit the object file. In the first example, /OBJECT is the default qualifier and need not be included explicitly. In other words, unless your MACRO command includes the /NOOBJECT qualifier, an object file is produced.

1.2.3 HELP

HELP is available from the system for all DCL commands and also for many other aspects of the system. You can get help through the HELP command or by typing a question mark (?) in response to any DCL prompt.

HOW TO USE DCL

For instance, if you wish help on the TYPE command, type the following:

```
$ HELP TYPE (RET)
TYPE[/qualifier[s]] filespec[s]
    /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
```

The TYPE command displays the contents of text files on your terminal.

\$

The HELP text consists of a brief explanation of the command followed by an illustration of the syntax, showing that TYPE accepts one or more file specifications, and one or more qualifiers. See Section 4.4.1 for more information on TYPE and its qualifiers. Or, type a HELP command naming the qualifier to get more information on that qualifier.

```
$ HELP TYPE TODAY
```

```
TYPE/TODAY filespec[s]
```

The /TODAY qualifier specifies that you wish the TYPE command to type only files created today.

If you want help while being prompted by the TYPE command, use the following procedure:

```
$ TYPE (RET)
File(s)? ? (RET)
TYPE[/qualifier[s]] filespec[s]
    /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
```

The TYPE command displays the contents of text files on your terminal.

```
File(s)? (RET)
```

The same help text is printed on your terminal, but the prompt returns, meaning the TYPE command is still waiting for you to list the files you want typed.

You can also get help on a specific subtopic while being prompted by a command by responding to the prompt with a question mark. For example, after getting help on SET, you can also get help on a specific function of SET:

```
$ SET (RET)
Function? ? (RET)
SET thing
```

HOW TO USE DCL

The SET command can be used to set something. The following things can be set with this command:

[DAY]TIME	DEBUG	DEFAULT	DEVICE
GROUPFLAGS	HOST	LIBRARY	[NO]PARTITION
PRIORITY	PROTECTION	QUEUE	SYSTEM
TERMINAL			

For information on the SET ACCOUNTING command, see HELP ACCOUNTING SET.

To get help on a specific topic, enter a question mark followed by the thing:

Function? ? DEFAULT (RET)

SET DEFAULT [ddnn:][[ufd]]

The SET DEFAULT command sets your default directory or device, or both.

SET DEFAULT with no parameters returns a nonprivileged user to the login device and UFD. For privileged users, the unadorned SET DEFAULT returns to the login device, but the UFD remains the same.

Function? (RET)

You can also get help by typing a question mark in response to the dollar sign prompt (\$) and to the explicit DCL prompt (DCL>).

If you should decide after reading the help text that you have chosen the wrong command, press CTRL/Z in response to the prompt to end the execution of the command. (A CTRL/Z in response to a prompt always cancels execution of a DCL command.)

From a DCL terminal, the command HELP provides information on DCL and the command HELP/MCR provides information on MCR. From an MCR terminal, the command HELP provides information on MCR and the command HELP/DCL provides information on DCL.

There may also be special help files providing information on special aspects of your installation. In addition, you can create local help files for your own use. See Chapter 3 for more information on the HELP command and help files.

1.2.4 MCR and DCL Commands

You can usually enter MCR commands from terminals set to DCL or DCL commands from terminals set to MCR.

If you are at a DCL terminal and want to enter an MCR command, simply precede the command with an MCR. The command is passed directly to MCR while the terminal remains set to DCL.

Here is the format of the MCR command:

DCL>MCR
MCR command line? mcrcommand

or

DCL>MCR mcrcommand

HOW TO USE DCL

And, if you want to enter a DCL command from an MCR terminal, use the DCL command. The command is executed and the terminal remains set to MCR. Here is the format of the DCL command:

```
MCR>DCL dclcommand
```

There are no prompts from this command.

1.2.5 Abbreviations

It is rarely necessary for you to type either the complete command name or the complete qualifier name. You only need to type the characters needed to distinguish the command or qualifier from all others.

For example:

- TYPE can be abbreviated T because it is the only command beginning with that character.
- INITIALIZE can be abbreviated INI, but not IN.
- INSTALL can be abbreviated INS, but not IN.

Three letters will usually be enough. Four letters will always be enough. You can often omit other parts of commands as well. You should experiment to find how short various commands can be cut. For instance, the following command:

```
$ SET TERMINAL/VT100 (RET)
```

is the documented format for the command that sets a terminal as a VT100. The same command to VAX/VMS DCL does the same thing. However, the RSX-11M version of DCL permits you to type the following:

```
$ SE VT100 (RET)
```

to achieve the same result. The second form does not work on VAX/VMS systems.

These briefer forms should be used interactively only, and not used when you are making a permanent record or creating an indirect command file. You should also see Section 1.7 for a discussion of compatibility between RSX-11M DCL and other versions of DCL used on other operating systems.

For your convenience, some frequently used commands have brief forms:

Command	Brief Form	Command	Brief Form
ABORT	A	HELP	H
BROADCAST	B	HELP	?
COPY	C	LOGOUT	LO
DIRECTORY	D	LINK	L
DEALLOCATE	DEAL	MACRO	M
DEASSIGN	DEAS	PRINT	P
EDIT	E	RUN	R
FORTRAN	F	SHOW	S

To save time and typing, use these brief forms to replace the command names when you are entering commands.

1.2.6 Numbers and Dates

DCL recognizes both octal and decimal numbers. You usually do not have to identify a number as octal or decimal. All numbers are recognized as being octal or decimal, unless otherwise noted.

DCL recognizes dates in two forms:

dd-mmm-yy as in 21-JUN-85

or

mm/dd/yy as in 6/21/85

System displays are always in the first format.

1.2.7 Multiple Parameters

Some commands permit you to enter a single file specification or other parameter by a list of parameters. If you are entering a list of parameters, each parameter must be set off by commas. For example:

```
$ PRINT ALPHA.TXT (RET)
```

causes a single file to be printed, while the following command:

```
$ PRINT ALPHA.TXT, BETA.TXT, GAMMA.TXT (RET)
```

causes three files to be printed. You have the option of including blanks on either side of the comma in lists.

If you end the list with a comma, DCL prompts you for further parameters. For instance:

```
$ PRINT ALPHA.TXT, BETA.TXT, (RET)
File(s)? GAMMA.TXT (RET)
```

Some commands accept a list of arguments to a single qualifier or parameter. In such cases, the list of arguments must be enclosed in parentheses, with each element set off by commas. For example, the following command:

```
$ MACRO/SHOW: (CALLS, COMMENTS) HIYA (RET)
```

If you need to enter only a single argument, you do not need the parentheses. For example:

```
$ MACRO/SHOW: CALLS HIYA (RET)
```

1.2.8 Underscore Character

The underscore character (_) is used to make DCL commands more readable where two words are needed to name a single command element, such as PRINT/FLAG_PAGE. However, you need not type the underscore to enter the command. PRINT/FLAGPAGE is the same as PRINT/FLAG_PAGE.

1.2.9 Colon and Equal Sign

The command descriptions in this manual show arguments set off by a colon (:), such as follows:

```
$ PRINT/COPIES:2 IZZY.TXT (RET)
```

You can always replace such colons with an equal sign (=), as in this example:

```
$ PRINT/COPIES=2 IZZY.TXT (RET)
```

Colons in device names, such as DB1:, and so forth, cannot be replaced by equal signs.

1.2.10 Quoting Strings

If you want to include an exact string in a DCL command, put the string in quotes. For instance, the following message:

```
$ BROADCAST/ALL Rock and roll will never die (RET)
```

is broadcast as follows:

```
11-MAY-85 13:55          From PRINCE::WRITERS (TT64:) to ALL
ROCK AND ROLL WILL NEVER DIE
```

while the following command:

```
$ BROADCAST/ALL "Rock and roll will never die" (RET)
```

is broadcast as follows:

```
11-MAY-85 13:56          From PRINCE::WRITERS (TT64:) to ALL
"Rock and roll will never die"
```

You also need quotes when passing commands to tasks using the /PARAMETERS qualifier to MOUNT or the /COMMAND qualifier to RUN or INSTALL.

1.2.11 ANSI File Names

ANSI file names do not have the same format as Files-11 file names and may contain characters that are illegal in Files-11. For instance, "MONEY&." is a legal ANSI file name, as is "CASH&.". To use those file names in a command, simply enter them as shown in the following example:

```
$ COPY (RET)
From? MM:"MONEY&." (RET)
To? MML:"CASH&." (RET)
```

If you are copying a file from a tape to a Files-11 disk, however, you must use only Radix-50 file specifications, such as the following:

```
$ COPY (RET)
From? MM:"MONEY&." (RET)
To? DB:[200,1]RADIX50.TXT (RET)
```

The Radix-50 character set consists of the numbers 0 through 9, the letters of the alphabet, the dollar sign (\$) and the period (.).

HOW TO USE DCL

1.2.12 DECnet and DCL

Some commands will accept DECnet node names as part of a file specification. They are as follows:

APPEND	CREATE	DIRECTORY	RENAME
COPY	DELETE	PRINT	TYPE

Several other DCL commands accept node names in the file specifications. However, DECnet modifies the basic syntax of these commands. These commands are: PRINT, SET PROTECTION, and SET FILE. Be sure to check the RSX DECnet Guide to User Utilities before attempting to use these commands.

If the node you select is part of your network, you can simply add the node to the file specification in the appropriate DCL command. Note that your terminal remains connected to your local node when you issue these commands to a remote node.

In the following example, you issue a command to type on your terminal the file ROMAN.TXT, which is located on remote node PRINCE and device DB2:, in directory [303,5].

```
$ TYPE PRINCE::DB2:[303,5]ROMAN.TXT (RET)
```

All the usual rules about file and volume protection are maintained, of course.

If the remote node you specify has a different style of file specification from RSX-11M, you must enclose the file specification in quotation marks, as shown:

```
$ TYPE EEETEE::"DISK$USERDISK:[DALTON]DESPERADO.DIS" (RET)
```

In addition to transmitting commands between nodes, DECnet also allows you to connect your terminal to a remote node. See the SET HOST command, described in the following section.

1.2.12.1 SET HOST

After you log in to a system, you can use the SET HOST command to connect your terminal to a different system.

Both your current system and the remote system must run DECnet software. In addition, you need to have an account on the remote system; otherwise, you will not be able to log in on the remote system after you issue the SET HOST command.

Format

```
SET HOST nodename
```

Parameters

nodename

Specifies the name of the remote system that you want to connect your terminal to.

A node is one system within a network of systems. The system that you originally log in on is called a local node; all other systems in the network are called remote nodes.

HOW TO USE DCL

Once you have connected to the remote node, that operating system responds with a prompt. After you log in, use commands that the remote operating system accepts.

Type the LOGOUT command to log out of a remote node. After typing this command, you are relocated to your local node.

You can use SET HOST to connect to only one remote host at a time. For example, suppose you want information located at two different remote nodes, KING and JUNE. You cannot connect first to remote node KING, and then execute SET HOST again to connect to remote node JUNE. You must first log out of KING, which relocates you to your local node, and then use SET HOST to connect to JUNE.

See the RSX-11 DECnet documentation accompanying the DECnet software product for a full explanation of this command.

Examples

```
$ SET HOST TOOTSI (RET)
```

```
Connected to node "TOOTSI", System type = RSX-11M  
System ID: RSX TIMESHARING
```

```
$ LOGIN (RET)
```

```
Account or Name: 303,3 (RET)
```

```
Password: (RET)
```

```
.  
. .  
.
```

```
$
```

This SET HOST command connects your terminal to the remote node TOOTSI. The remote system identifies itself, then prompts you. You log in on the remote system by using the name and password of your account on that system.

1.2.12.2 SHOW HOST

The SHOW HOST command displays the name of the processor to which your terminal is connected currently. The display also shows you the name and version number of the operating system running on the processor.

The SHOW HOST command is most useful after you have connected your terminal to a remote system with the SET HOST command. However, SHOW HOST works whether or not your system runs DECnet software. Without DECnet on your system, this command simply displays information about your local operating system.

Format

```
SHOW HOST
```

Example

```
$ SHOW HOST (RET)
```

```
HOST=TOOTSI RSX-11M V4.2 BL38
```


HOW TO USE DCL

This example indicates the display from SHOW HOST. The name of your current processor is TOOTSI, which is running Version 4.2 of the RSX-11M operating system.

1.2.13 Command Line Continuation

Sometimes a command will not fit all on one line. In that case, you can continue the line by using the hyphen (-). When you end a command line with a hyphen and a carriage return, the DCL continuation prompt (->) indicates that you can continue entering the command line. If you are continuing a line from a prompt, such as Task? that prompt is the indication that the line is being continued.

This feature permits you to enter command lines including more characters than your terminal has room for on one line.

DCL commands are limited to 80 characters in all. When you type a continuation line, count the hyphen and the two times you press the RETURN key as three of the 80 characters. Of course, each blank, as well as each punctuation mark, counts as one character.

Here is an example of line continuation:

```
$ PRINT/COPIES:2/LENGTH:60/FLAGPAGE/AFTER:(04-JUL-85) OZY.TXT,- (RET)
->IZZY.TXT, FIZZY.TXT (RET)
```

The command is not entered until DCL encounters a line ending with a RETURN that is not preceded by a hyphen. In the example, the first carriage RETURN does not enter the command. Only the second RETURN enters the command. The RETURN can be on a line by itself.

1.2.14 Comments in Command Lines

You may want to include comments in command lines if you are keeping a permanent record, such as a command file, or simply an interactive session on a hardcopy terminal. You can include comments in a DCL command line with the exclamation point (!).

If the comment ends the command line, only a single exclamation point is needed, as in this example:

```
$ PRINT OZY.TXT !Poem by Shelley (RET)
```

If the comment is within the command line, two exclamation points are needed to set it off, as in this example:

```
$ PRINT!Parody of Shelley! IZZY.TXT (RET)
```

These comments are ignored and not interpreted in any way by DCL.

Comments can be placed at any natural break in the command line: between qualifiers, between parameters, even as part of a response to a prompt. Another example:

```
$ PRINT/FORMS:2!Letter-quality printer! IZZY.TXT (RET)
```

HOW TO USE DCL

1.2.15 SET DEBUG

SET DEBUG displays the MCR translation of any DCL command.

Format

```
SET [NO]DEBUG[/qualifier[s]]
```

Command Qualifiers

```
/[NO]EXECUTE  
/FULL
```

Command Qualifiers

```
/[NO]EXECUTE
```

Indicates that you wish the DCL command to execute after the translation is displayed. The default is /NOEXECUTE, meaning that the translation only is displayed. When SET DEBUG/NOEXECUTE is in effect, you can only issue SET DEBUG commands or MCR commands using the DCL MCR command; all other DCL commands are trapped and translated. You can issue SET NODEBUG to turn off SET DEBUG. You can also issue SET DEBUG with either /FULL or /EXECUTE or both.

```
/FULL
```

Indicates that you wish SET DEBUG to display the logical symbols (IFORM queue) used by DCL to translate the command as well as the translation itself. See the RSX-11M/M-PLUS System Management Guide for information on how DCL uses these logical symbols.

Examples

```
$ SET DEBUG (RET)  
$ SHOW TIME (RET)  
TIM  
$ INITIALIZE/EXTENSION:200/PROTECTION:(SYS:RWED,OWN:RWED,GRO,WO)(RET)  
Device? DK2 (RET)  
Label? HOTROD (RET)  
INI DK2:HOTROD/EXT=200./PRO=[RWED,RWED,,]  
$ SET DEBUG/EXECUTE (RET)  
$ SH TIM (RET)  
TIM  
21:33:46 11-FEB-85  
$ SET NODEBUG (RET)  
$
```

In this example, the user first issued SET DEBUG. Then the user typed SHOW TIME. DCL displayed TIM, which is the MCR command to display the date and time, but no date and time is displayed because /NOEXECUTE was in effect. Then the user typed an INITIALIZE command and DCL displayed the MCR translation of the same command. Then the user typed SET DEBUG/EXECUTE and typed SHOW TIME again. This time, after displaying the MCR command TIM, DCL passed the command TIM to MCR and the date and time were displayed. Finally, the user typed SET NODEBUG to end session with SET DEBUG.

Notes

Use SET DEBUG to check on the MCR syntax of any DCL command. This is particularly helpful when learning the brief forms of commands.

HOW TO USE DCL

SET DEBUG creates your own copy of DCL that remains present while SET DEBUG is in effect. Therefore, SET DEBUG creates a load on the system. You should use SET DEBUG sparingly, particularly when the system is under heavy use.

1.2.16 Errors

You can correct typing errors or change the line completely, by using the RUBOUT or DELETE key, or by pressing CTRL/U (provided you have not terminated the line).

You can get rid of any DCL command by pressing a CTRL/Z, provided you have not terminated the command.

If the system detects an error in the command line input, it returns the appropriate error message at the issuing terminal.

Here are some examples of incorrect commands and the error messages they produce:

```
$ PRIJT IZZY.TXT (RET)
DCL -- Illegal command
```

```
$ PRINT/PURPLE IZZY.TXT (RET)
PRINT -- Illegal or contradictory qualifier
PRINT/PURPLE IZZY.TXT
^
```

```
$ PRINT/COPIES:TWELVE IZZY.TXT (RET)
PRINT -- Numeral expected
PRINT/COPIES:TWELVE IZZY.TXT
^
```

In the first case, the error was detected by DCL, as indicated by the first part of the error message. There is no DCL PRIJT command. The entire command was rejected.

In the second case, the command was entered correctly, but the qualifier was incorrect. The first part of the message shows that the error was detected within the PRINT command itself. The command is reprinted and a circumflex (^) points to the error.

In the third case, the command and qualifier were correct, but the argument was in error. The message explains the error and the circumflex points to the error.

Sometimes the circumflex does not point directly at the error, but at the point at which the command started to go wrong, which may be several characters before or after the actual error. Typing mistakes are by far the most common cause of errors. Retyping the command often eliminates the error. Other common causes of errors are omitting a space or other delimiter in a command line, specifying invalid devices or nonexistent files, issuing privileged commands from a nonprivileged terminal, and failing to type a sufficient number of characters to distinguish the command or command element.

The command descriptions include the most common errors produced by the commands and suggestions for correcting the errors. All the DCL error messages are listed and explained in Appendix A of this manual.

1.3 FUNCTIONAL GROUPING OF DCL COMMANDS

This manual includes seven functional groupings of commands, each grouping constituting a chapter. Each of these chapters is described here. Fully detailed descriptions of each command are included in these functional chapters. The functional groupings of commands are described in Sections 1.3.1 through 1.3.7. Each functional chapter also includes an introduction to the important concepts of the functional group.

Command descriptions can have five parts:

1. **Brief statement of the function of the command.**
2. **Full description of the format of the command, including prompts, defaults, acceptable values for arguments, and the effects of each qualifier.**
3. **Examples of the command in use.** These examples often illustrate less obvious aspects of the use of the command.
4. **Notes, including warnings about side effects, counteracting commands, recommendations for further reading, and so forth.** Some command descriptions do not have notes.
5. **Error messages, including an explanation and a suggested user action for each error.** Only error messages specific to the command are included in the command descriptions. General error messages, such as those relating to syntax, are described in Appendix A.

In addition, the formats of all RSX-11M DCL commands are presented in alphabetical listing in Chapter 9.

Nonprivileged commands are those commands needed in everyday use of the system by all users. Many privileged commands are also included in DCL. Privileged commands are those commands that affect system operations. For instance, the SHOW TIME command is a nonprivileged command, but the SET TIME command is privileged.

1.3.1 The SET and SHOW Commands

The SET commands dynamically alter system characteristics. The SHOW commands display system characteristics. These commands are introduced in Chapter 2. The most useful functions of SET and SHOW are also described in Chapter 2. However, because such a wide variety of functions can be altered or displayed, the remaining functions of SET and SHOW are listed in Chapter 2 but described in the appropriate functional chapter. SET and SHOW are the last commands described in each chapter. For example, SET DEVICES and SHOW DEVICES are described in full at the end of Chapter 5; and SET TERMINAL and SHOW TERMINAL are described at the end of Chapter 3.

The SET or SHOW command and its function are considered to form a single command. That is, rather than referring to the TERMINAL function of the SHOW command, you would refer to the SHOW TERMINAL command.

1.3.2 Terminal Operations

All commands directly related to terminal operations are described in Chapter 3. The commands include BROADCAST, HELP, LOGIN, LOGOUT, REQUEST, SET TERMINAL, and SHOW TERMINAL. In addition, Chapter 3 includes a detailed description of the use of the terminal keyboard on RSX-11M systems. All control (CTRL) characters and other important keys are explained.

1.3.3 Handling Files

Most information in RSX-11M systems is in files in User File Directories (UFDs). These files and directories are located within Files-11 volumes.

All commands relating to files are described in Chapter 4.

The commands include ANALYZE/MEDIA, APPEND, CONVERT, COPY, CREATE, CREATE/DIRECTORY, DELETE, DIFFERENCES, DIRECTORY, EDIT, EDIT/EDT, EDIT/KED, EDIT/SLP, PRINT, PURGE, RENAME, SORT, TYPE, and UNLOCK.

SET and SHOW commands described in Chapter 4 include SET and SHOW DEFAULT, SET FILE, and SET PROTECTION.

1.3.4 Handling Devices and Volumes

Volumes are held on magnetic media, usually disks. The volume is in the RSX-11M format, called Files-11. The disks are mounted on hardware devices. Devices can be as follows:

- Public - available to all users
- Shareable - available to some users but not others
- Private - available to a single user only

All commands relating to volumes and devices are described in Chapter 5. The commands include ALLOCATE, ASSIGN, ASSIGN/REDIRECT, ASSIGN/TASK, BACKUP, DEALLOCATE, DEASSIGN, DISMOUNT, INITIALIZE, INITIALIZE/UPDATE, and MOUNT.

SET and SHOW commands described in Chapter 5 include SHOW ASSIGNMENTS and SET and SHOW DEVICES.

1.3.5 Program Development

RSX-11M provides full facilities for program development, which are described in Chapter 6. The commands described include LIBRARY and LINK.

SET GROUPFLAGS and SHOW GROUPFLAGS are also described in Chapter 6.

HOW TO USE DCL

1.3.6 Running Tasks

The task is the fundamental executable programming unit on RSX-11M. All commands related to running tasks are described in Chapter 7.

These commands include ABORT, CANCEL, CONTINUE, FIX, INSTALL, REMOVE, RUN, START, STOP/BLOCK, START/UNBLOCK, and UNFIX.

SET and SHOW commands described in Chapter 7 include SHOW COMMON, SHOW CLOCK_QUEUE, SET and SHOW PARTITION, SET PRIORITY, and SHOW TASKS.

1.3.7 System Control

System control commands are described in Chapter 8. Most of these commands are privileged, but the SHOW commands described in Chapter 8 are nonprivileged.

Only SET and SHOW commands are described in Chapter 8. These include SET and SHOW [DAY]TIME, SHOW MEMORY, SET and SHOW SYSTEM/DIRECTORY, SET SYSTEM/[NO]LOGINS, SET and SHOW SYSTEM/POOL, and SHOW USERS.

1.4 USING THE QUEUE MANAGER: PRINT JOBS

The RSX-11M Queue Manager provides facilities for printing files on line printers or other output devices. Files can be printed under user control or under the control of a system task or applications task. See the description of the PRINT command in Section 4.4.2.

All uses of the Queue Manager are described in a separate manual, the RSX-11M/M-PLUS Batch and Queue Operations Manual. The manual introduces and explains all the uses of the Queue Manager. The manual includes sample print jobs, all commands for placing jobs in queues and displaying queue contents, and a complete description of the process of setting up the Queue Manager. In addition, the manual includes an explanation of writing tasks to spool output.

1.5 INDIRECT COMMAND FILES IN DCL

In addition to the batch-processing subsystem, RSX-11M provides the Indirect Command Processor (Indirect) as an alternative means of automatically passing commands to the operating system. If you have a series of commands to be executed in the same or similar fashion every time, you can include these commands in a file to be run by Indirect.

Indirect accepts not only DCL and MCR commands, but also special directives that allow you to program the execution of the indirect command file. Indirect is described in full in the RSX-11M/M-PLUS Indirect Command Processor Manual.

An indirect command file can include both MCR and DCL commands. If the file contains commands to both of these command line interpreters (CLIs), you must either prefix each CLI command with the CLI name, or change the CLI setting of your terminal before each CLI command. When your file has completed, reset the terminal to your usual CLI in the indirect command file. For example, if your terminal is set to DCL and you have MCR commands in your indirect command file, you can prefix each MCR command with "MCR," or set the terminal to MCR before each command and then reset it to DCL afterwards.

HOW TO USE DCL

1.6 DCL AND RSX-11M UTILITIES AND SYSTEM TASKS

The RSX-11M operating system includes a number of utilities and other system tasks designed to assist you in your use of the system. Most utilities are used to create or alter files in some way.

Many DCL commands invoke these utilities transparently to the user. Just as some DCL commands are passed to MCR, other DCL commands are passed to utilities.

One important utility is the Peripheral Interchange Program (PIP), which is used to manipulate files. DCL functions such as APPEND, COPY, DELETE, PURGE, and DIRECTORY depend on PIP. As another example, the LINK command invokes the RSX-11M Task Builder. Appendix B of this manual gives information on the relationships between DCL and the utilities and other system tasks.

For more information on each RSX-11M utility, see the RSX-11M/M-PLUS Utilities Manual. RMS-11 utilities are described in the RMS-11 documentation supplied with your system. Some system tasks that might be considered utilities, such as the Task Builder, are described in separate manuals.

For descriptions of other useful and important system tasks, see the following manuals:

- RSX-11M System Generation and Installation Guide - a description of the process of tailoring a system to a particular installation.
- RSX-11M/M-PLUS MCR Operations Manual - a description of MCR plus many details on system startup procedures and system operations.
- RSX-11M/M-PLUS Indirect Command Processor Manual - a full description of the Indirect Command Processor.
- RSX-11M/M-PLUS Batch and Queue Operations Manual - a description of the Queue Manager, which controls the orderly printing of files and other output, and the batch processing subsystem, a means of automatic use of the system. Batch processing is available on RSX-11M-PLUS systems only.
- EDT Editor Manual - describes the use of EDT, the DEC Standard editor. The manual includes an introduction to EDT.
- RSX-11M/M-PLUS Guide to Program Development - a detailed introduction to the program development facilities, both DCL and MCR, of RSX-11M/M-PLUS.
- PDP-11 MACRO-11 Language Reference Manual - an introduction to the MACRO-11 Assembly Language and Assembler.
- RSX-11M/M-PLUS and Micro/RSX Task Builder Manual - a description of the task building, or linking, process in RSX-11M.

HOW TO USE DCL

1.7 RSX-11M DCL AND VAX/VMS DCL

DCL on RSX-11M Version 4.2 and subsequent releases is designed to be compatible with DIGITAL standard DCL on VAX/VMS Version 3.0 and subsequent releases. If the function being performed is identical, the same DCL command will have the same effect on an RSX-11M system or a VAX/VMS system. As other operating systems adopt DCL, the same will be true for them.

In most cases, the default forms of DCL commands will have the same effect on either kind of system. Furthermore, many qualifiers are the same and also have the same effect.

On the other hand, operating systems that implement DCL, including RSX-11M, will have system-specific extensions to DCL. For example, RSX-11M systems have several features not available on VMS systems; DCL commands have been implemented to invoke these features. The command syntax for these features is different from any VMS command syntax to avoid confusion.

In addition, many of the features discussed in this chapter are RSX-11M extensions to the basic DCL language. Prompts, HELP, error messages, system displays, and command parsing (including abbreviating) are unique to RSX-11M and may not be fully compatible with other DCL implementations.

NOTE

All the short forms and abbreviations in DCL are intended for the convenience of interactive users at terminals. You should not use either these brief command forms, or other abbreviations shorter than four characters per command element, in indirect command files or any other permanent form. As new commands are added to DCL, these abbreviations may change, but the full-format commands are stable.

CHAPTER 2

THE SET AND SHOW COMMANDS

The SET commands dynamically alter system characteristics. The SHOW commands display system characteristics.

These commands affect every part of the operating system. Some of them are needed by every user every day. Others might not be used once in a year.

SET and SHOW both prompt

Function?

There are more than 20 functions that can be altered or established with the SET command, and more than 20 others, not all of them the same, that can be displayed with the SHOW command. Most of the SET functions include options, values, or negations.

Customarily, the SET or SHOW command and the function are considered to form a single command. That is, rather than speaking of the TERMINAL function of the SHOW command, you would refer to the SHOW TERMINAL command.

Some of the most commonly used variations on the SET and SHOW commands are described for your convenience in Sections 2.1 through 2.12.

Full descriptions of all the SET and SHOW command functions appear later in this manual. Each chapter concludes with the SET and SHOW commands appropriate to that chapter. Thus, SET and SHOW TERMINAL are described at the end of Chapter 3, Terminal Operations, and SET and SHOW DEVICES are described at the end of Chapter 5, Handling Devices and Volumes. See the Table of Contents for references.

The following are the most commonly used SET and SHOW commands, described here in brief form for your convenience. All of these commands are described in detail elsewhere in this manual.

2.1 SET AND SHOW TIME

Format

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

You can set both the time and the date with this privileged command. You can enter the date in either form, for example, 04-JUL-85, or 7/4/85. In the latter case, mm stands for month number. No matter how you enter the date, it will be displayed in the first form.

See Section 8.1.1.

THE SET AND SHOW COMMANDS

The SHOW [DAY]TIME command displays the current setting of the day and time on your system.

Format

```
SHOW [DAY]TIME
```

See Section 8.1.2.

2.2 SHOW MEMORY

```
SHOW MEMORY
```

This command brings up the RMD display, which provides information on task activity, partitions, drivers, CPU usage, and so forth on your system.

Format

```
SHOW MEMORY
```

On video terminals, this display is dynamic. On hardcopy terminals, the command prints a snapshot of the current RMD display.

The display provides a rough picture of the current state of the system. Alternate displays include the Active Task List or a single task header.

Not all systems include this command for interactive use. Most systems, however, do run the display on a slaved video terminal somewhere near the computer.

See Section 8.1.7 for more information on the memory display. See Section 7.12.4 for more information on the task list and task header displays.

2.3 SET AND SHOW TERMINAL

You can set and display many attributes of your terminal. The command

Format

```
SHOW TERMINAL
```

without any qualifier, displays the current setting of all settable attributes of the terminal, as shown:

```
TT5:  [303,5]          [303,5]
      CLI = DCL      BUF = 132.    HFILL = 0      SPEED=(300:300)
      LINES = 66.    TERM = LA34    OWNER = NONE   BRO      NOABAUD
      LOWER  NOPRIV  NOHOLD  NOSLAVE NOESC  NOCRT   HFF      NOREMOTE
      ECHO   NOVFill NOHHT  NOFDX  WRAP   NORPA  NOEBC   TYPEAHEAD
      AVO   ANSI   DEC    NOEDIT  NREGIS NOSOFT NOBLKMOD
      NOHSYNC  NOPASTHRU  TTSYNC  NOPRINTER_PORT
```

THE SET AND SHOW COMMANDS

The display is explained in full in Section 3.8.2. Here, briefly, are the more significant fields:

Terminal number, login UIC, default UFD

CLI	Current CLI setting
BUF	Buffer-length, that is, the width in columns
SPEED	Transmit and receive speed
LINES	Page length in lines
TERM	Terminal model
OWNER	Tells whether terminal has been allocated
BRO	Terminal can receive broadcast messages
NOPRIV	Terminal is not privileged

The SET TERMINAL command permits you to set most of these attributes for your terminal. Privileged users can set the attributes of any terminal.

Format

SET TERMINAL/attribute[/attribute[s]]

Note that you can set more than one attribute at a time. A command such as the following:

```
$ SET TERMINAL/VT100/LOWERCASE/DCL (RET)
```

is accepted by the system and executed.

Here is a summary of the most important attributes you can set:

/DCL	/MCR	/UPPERCASE	/LOWERCASE
/PAGE_LENGTH:n	/WIDTH	/SPEED:(transmit, receive)	
/[NO]BROADCAST	/[NO]PRIVILEGE	/SCOPE	/HARD_COPY
/CLI=cliname	/HOLD_SCREEN	/INQUIRE	

You can also set your terminal as a particular model:

/ASR33	/LA36	/LQP02	/VT61
/ASR35	/LA38	/LQP03	/VT100
/DTC01	/LA50	/RT02C	/VT101
/KSR33	/LA100		/VT102
/LA12	/LA120	/VT05B	/VT105
/LA30P	/LA180S	/VT50	/VT125
/LA30S	/LA210	/VT52	/VT131
/LA34	/LN03	/VT55	/VT132
/VT200_SERIES	/PRO_SERIES		

See Section 3.8.1 for more information on the meaning of these attributes.

2.4 SHOW DEVICES

The SHOW DEVICES command displays the name and other information for all devices known to the system or, if specified, information for a specific device.

Format

SHOW DEVICES

THE SET AND SHOW COMMANDS

This command displays the name and other information for all devices of a particular type known to the system, where dd is the 2-letter mnemonic for the device type or pseudo-device type. See Section 5.11.3.

2.5 SHOW QUEUE

This command displays information about the queue file maintained by the Queue Manager. Two forms of this command are most useful.

Format

SHOW QUEUE

This command displays information about all jobs in all queues. In addition to showing all queue assignments, the information display includes all Queue Manager QMG jobs and all files included in those jobs.

SHOW QUEUE/FULL

This command displays full information about all jobs in all queues. In addition to the information displayed by SHOW QUEUE, SHOW QUEUE/FULL displays all the attributes of all the QMG jobs in the queue, such as /FORMS, /COPIES, and /PAGE_LENGTH.

See the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information on the SHOW QUEUE commands.

2.5.1 SHOW PROCESSOR

The SHOW PROCESSOR command displays information about the card readers, batch processors, printers, and other devices under the control of the Queue Manager.

Format

SHOW processor_{type} processor_{name}[/qualifier[s]]

Parameters

processor_{type}

The processor_{type} is one of the following:

BATCH
CARD READER
DEVICE
INPUT
PRINTER
PROCESSOR

CARD_READER and INPUT are synonyms. DEVICE refers to all nonbatch output processors and is a synonym for PRINTER. BATCH refers to all batch processors.

A simple SHOW PROCESSOR command displays all processors in the system.

See the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information.

THE SET AND SHOW COMMANDS

processorname

For physical devices, the processor name is identical with the name of the physical device. Thus, the device LPl: is controlled by processor LPl:.

Applications processors have names of six characters.

2.6 SHOW ASSIGNMENTS

The SHOW ASSIGNMENTS command displays at your terminal all local and login logical-device assignments. See Section 5.11.1.

Format

```
SHOW ASSIGNMENTS[:ttnn:]
```

2.7 SET AND SHOW DEFAULTS

The SET DEFAULT command alters your default settings for device and directory.

Format

```
SET DEFAULT [ddnn:][g,m]
```

Parameters

dd

The 2-letter device mnemonic

nn

The 1- or 2-digit device number terminated by a colon

[g,m]

The directory is two octal numbers separated by a comma and enclosed in brackets.

The default device name and directory are included automatically in every file specification included in a command unless overridden. See Section 4.5.1.

The SHOW DEFAULT command displays your current default device and directory. See Section 4.5.2.

Format

```
SHOW DEFAULT
```

2.8 SHOW TASKS

This command displays information about tasks in the system. Four forms of the command are most useful.

The SHOW/TASKS/ACTIVE command displays information about the tasks active at your terminal.

THE SET AND SHOW COMMANDS

Format

SHOW TASKS/ACTIVE

The SHOW TASKS/ACTIVE/ALL command displays information about all tasks active on the system.

Format

SHOW TASKS/ACTIVE/ALL

The SHOW TASKS/INSTALLED command displays full information about all tasks installed on the system.

Format

SHOW TASKS/INSTALLED

The SHOW TASKS/INSTALLED/DEVICE command displays full information about all tasks installed on the specified device.

Format

SHOW TASKS/INSTALLED/DEVICE:ddnn:

The following commands display full information on a single active or installed task.

Format

SHOW TASKS/ACTIVE FULL taskname

SHOW TASKS/INSTALLED FULL taskname

See Section 7.12.3.

2.9 SHOW USERS

The SHOW USERS command displays the terminal number and login UIC for each logged-in terminal.

Format

SHOW USERS

See Section 8.1.6 for more information.

2.10 SET PROTECTION

The SET PROTECTION command alters the access protection of a file or files. If you wish to protect more than one file, the file specifications must be separated by commas.

Format

SET PROTECTION filespec[s]
Protection? (SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:R)

THE SET AND SHOW COMMANDS

Parameter

filespec

The filespec parameter uniquely identifies a file. The file specification specifies the device name on which the volume containing the file is mounted, the directory in which the file is located, the file name, the file type, and the version number of the file.

There are four kinds of access to files and four classes of users who may access files. The previous example shows the code for the default protection used on the system.

The four classes of users who may access files are as follows:

- SYSTEM Privileged users and the operating system itself.
- OWNER Users logged in under the same User Identification Code (UIC) the file was created under.
- GROUP Users logged in under the same group number as the UIC file was created under.
- WORLD All other users

The four kinds of access to files are as follows:

- R READ access. The file can be typed, printed, copied, run (if a task image), and so on.
- W WRITE access. The file can be written to or altered.
- E EXTEND access. The file size can be changed.
- D DELETE access. The file can be deleted.

The access you specify is the access that is given. If you specify READ access to users in a particular category, those users will be able to type, print, copy, or run (if it is a task image) your files. However, your files are protected from being edited or deleted by those users.

The protection status shown at the beginning of this section is the default protection status accorded every file on the system. Use SET PROTECTION to change the protection status of files.

You must enter the class of user and the type of access in the form shown. If you do not wish to change the access rights of a particular class of user, simply leave that class out.

All users can set the protection of their own files. Privileged users can set the protection of any file. Note that you can set the protection of a file so no user, including yourself, can do anything with the file. If you do this, you will need a privileged user to reset the protection when you wish to use the file.

You can display the protection of a file with the DIRECTORY command and /FULL qualifier. See Section 4.3.1.

THE SET AND SHOW COMMANDS

2.11 SET AND SHOW SYSTEM/DIRECTORY

The Executive, privileged system tasks, and other system files are kept together in a single directory called the system directory. In most cases, this will be the directory [3,54] on the system disk. An installation may, however, have more than one UFD containing such files, and, therefore, more than one system. For example, an installation may have one system for regular use and a second for backup or testing. The two systems can have quite different characteristics.

The SET SYSTEM/DIRECTORY privileged command sets the system directory as follows:

Format

```
SET SYSTEM/DIRECTORY:[g,m]
```

The SHOW SYSTEM/DIRECTORY command displays the current system directory.

Format

```
SHOW SYSTEM/DIRECTORY
```

On RSX-11M systems, the system directory also includes the utilities and other nonprivileged tasks.

The system directory on RSX-11M systems makes possible the RUN \$taskname convention. When you issue a RUN or INSTALL command preceded by the dollar sign (\$), you are instructing the system to look in the system UFD for a task image file of the name you have given, and to run or install the task from that directory. This convention saves you from having to find the files and specify the tasks by device and directory.

See Sections 8.1.3 through 8.1.5 for more information on these directories and their use.

2.12 SET DEBUG

The SET DEBUG command allows you to see the MCR translation of any DCL command.

Format

```
SET [NO]DEBUG[/[NO]EXECUTE]
```

The default is /NOEXECUTE. This means that when you type a DCL command, the MCR translation appears on your terminal but the command is not executed. If you specify /EXECUTE, the translation appears on your terminal before the command is executed. See Section 1.2.15 for more information.

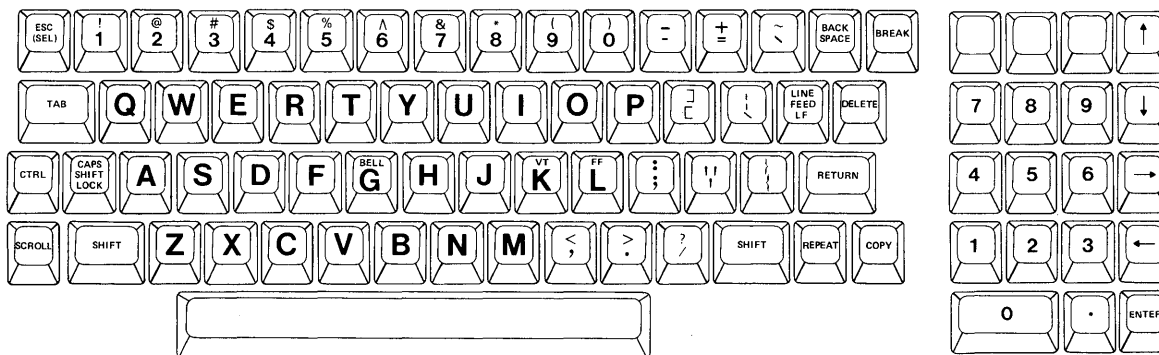
CHAPTER 3

TERMINAL OPERATIONS

The terminal is your main channel of communication with the computer system. The information in this chapter tells you how to use a terminal to communicate and interface with the RSX-11M operating system. For a quick review of basic operations at a terminal, refer to the Introduction to RSX-11M.

The discussions in this chapter are organized around three DIGITAL terminals: the LA36 hardcopy terminal (DECwriter II) and the VT52, VT100, and VT200-series video display terminals. These terminals do not always behave in exactly the same way; variations are noted in the descriptions of terminal functions in this chapter. If you are using another type of terminal, check with your system manager for special instructions. In any case, you should read the manual shipped with your terminal to get full use of your terminal on the system.

A typical terminal keyboard is shown in Figure 3-1. A keyboard is a set of alphanumeric keys, similar to the set on a typewriter. The keypad shown in the illustration is a set of special keys. Where pertinent, the functions of these keys will be explained in this chapter. All terminals will have some form of a keyboard; some terminals will not have the keypad.



ZK-1157-82

Figure 3-1 Terminal Keyboard and Keypad

Each terminal on an RSX-11M system has a number. Your terminal number is displayed when you log in. Your terminal number is also used by the system to identify tasks run from your terminal. (See Section 7.1.1 for more information on task naming.)

TERMINAL OPERATIONS

You can set many characteristics for your terminal using the SET TERMINAL command (Section 3.8.1). The characteristics of your terminal can be displayed with the SHOW TERMINAL command (Section 3.8.2). In addition, the SHOW USERS command (Section 8.1.6) displays the terminal number, default directory, and login UIC for each logged-in user.

Frequently, users work from more than one terminal. You can always use the pseudo-device name TI: to refer to the terminal you are currently using. You do not need a specific number. In most cases, when a system task requires you to name an output file, you can specify TI: and the output will be printed on your terminal.

A terminal is said to be attached if all its input and output is directed to or from a task. If you invoke a task, it may attach your terminal so that no other task can use the terminal for I/O. When you exit (for example by pressing CTRL/Z), the task detaches the terminal. When you are using an editor, the editor has your terminal attached.

Terminals may also be dedicated exclusively to a task. Such terminals are called slaved terminals. Tasks can slave terminals, or terminals can be set slaved through the SET TERMINAL command. Terminals can also be set slaved on logging in. Slaved terminals can communicate only with tasks soliciting input from the terminal.

3.1 LOGGING IN AND LOGGING OUT

Your terminal must be connected to the system hardware and the system software before you can use it. You can assume it is connected to the hardware. Connecting to the software is called logging in; disconnecting from the software is called logging out.

Logging in informs the system that you will be using a terminal. Logging out informs the system that you are through using a terminal.

Most terminals have an accessible on-off switch. This switch supplies power to the terminal, but it has no bearing on whether the terminal is logged in. If you turn the power switch off, a logged-in terminal remains logged in.

You can test whether a terminal is turned on and available by pressing RETURN. The RETURN key causes either the right-angle-bracket prompt (>) or the dollar sign (\$) to appear if your terminal is set to MCR or DCL, respectively.

The SHOW DEFAULTS command tells you whether the terminal is logged in or not. If it is not logged in, the command produces an error message. If it is logged in, the command displays the current default device and directory.

If the terminal is already logged in, you should type RETURN to determine whether the terminal is set to DCL or MCR. If the terminal is set to MCR, type in ACT as shown:

```
>ACT (RET)
```

If the terminal is set to DCL, type in SHOW TASKS/ACTIVE as shown:

```
$ SHOW TASKS/ACTIVE (RET)
```

TERMINAL OPERATIONS

This command will inform you whether the terminal is in active use or not. It is not good practice to take over a terminal while another user has tasks (other than CLIs) active at the terminal, as you may interfere with the other user's work. Check also with SHOW DEVICES to see if the other user has devices allocated or volumes mounted. If the other user has no significant activity under way, use your own judgment as to whether you should log the other user out, by typing the following:

```
$ LOGOUT (RET)
```

If your terminal is set to MCR, execute the following command to log out:

```
>BYE (RET)
```

Then, after the terminal is logged out, log yourself in.

If you are not sure how to log in, type HELP for help in logging in.

If the terminal is not logged in, you may log in. Terminals that are not logged in are not set to either DCL or MCR. You can log in to an RSX-11M multiuser protection system using either HELLO or LOGIN together with your User Identification Code and your password. The system's Account File Maintenance Program (ACNT) identifies each user as either a DCL or MCR user and sets the terminal accordingly. See the description of LOGIN (Section 3.1.1) for further information on the effects of logging in.

Remember, you can change the CLI of your terminal from DCL to MCR (or the reverse) with the following commands:

```
>SET /DCL=TI: (RET)
```

or

```
$ SET TERMINAL/MCR (RET)
```

All RSX-11M systems with DCL are multiuser protection systems. This means that the system includes features, such as LOGIN and LOGOUT and ALLOCATE (Section 5.5), that enable many users to use the system without interfering with each other's work.

LOGIN

3.1.1 LOGIN

LOGIN grants access from a terminal to a multiuser protection system. LOGIN also establishes certain characteristics of your terminal session.

Format

```
LOG[IN]
userid
Password: password
```

```
LOGIN userid
PASSWORD: password
```

```
LOGIN uic/password
```

Parameters

userid

Specifies the User Identification Code (UIC) of the user logging in. Five forms of UIC are accepted by this command:

```
[g,m]
g,m
[g/m]
g/m
name
```

Each user had a unique UIC, which the system manager assigns when setting up your account. The g is your group number and the m is your member number.

If you log in with a comma in the UIC, or if you log in by name, a file called LB0:[1,2]LOGIN.TXT is printed on your terminal. This file is prepared by your system manager and usually contains information about the system and other announcements. If your system displays such messages, using a slash (/) instead of a comma (,) between the group and member numbers suppresses these messages.

Your system may also have a file called LB0:[1,2]SYSLOGIN.CMD. This file can contain a number of system-level commands that your system manager wishes to have executed each time a user logs in.

Finally, you, as an individual user, can prepare a file called LOGIN.CMD to be kept in your UFD (User File Directory) to be executed each time you log in. This LOGIN.CMD file usually includes commands for setting up the terminal and the account for regular use.

The LOGIN command interacts with the Account File Maintenance Program (ACNT) to establish a number of characteristics of the terminal session you are beginning. These include your default device (where your files are located), and an indication of whether your account is privileged or not. See your system manager for information on how this will affect you.

LOGIN (Cont.)

password

Your password is limited to 6 Radix-50 characters, including dollar signs (\$), periods (.), exclamation points (!), quotation marks ("), and hyphens (-). If you enter your password in response to the **Password:** prompt, your password is not echoed on the screen. If you type in your password on the same line as your UIC, the password is echoed. Your password is established by your system manager as part of setting up your account.

Examples

```
$ LOGIN (RET)
Account or name: KAFKA (RET)
Password: (RET)

RSX-11M      V4.2 BL38 Multiuser [3,54] System
25-May-1985 09:16  Logged on Terminal TT54:

Good Morning

May 25, 1985                System Manager

** System will be down tonight from 21:00 to 24:00 **

                ** DISK-TO-DISK BACKUPS **

Please purge your files!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

May 16, 1985                Operator

Magtape drive MM1: is down until further notice.

$
```

In this example, full login text is printed automatically on the terminal because the user logged in with a name (KAFKA). The full login text is also printed if you log in with a UIC entered with a comma [g,m]. See next example.

The prompt signifies the completion of logging in.

```
$ LOGIN 303/5 (RET)
Password: (RET)

RSX-11 V4.2 BL38 Multiuser [3,54] System
25-MAY-85 09:18  Logged on Terminal TT56:

Good Morning

May 20, 1985                System Manager

** System will be down tonight from 21:00 to 24:00 **

                ** DISK-TO-DISK BACKUPS **

Please purge your files!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$
```

LOGIN (Cont.)

This example shows the first login of the day for this user. The UIC was entered with a slash (303/5). The printing of the system LOGIN.TXT file in full was, therefore, suppressed. Only the first message (defined as being a priority message by the system manager) was printed. The priority message is printed only on the first login of the day. This ensures that users who habitually log in with the slash do not miss anything. See the next example.

```
$ LOGIN 303/5 (RET)
Password: (RET)

RSX-11M V4.2 BL38 Multiuser [3,54] System
25-MAY-85 10:36 Logged on Terminal TT54:

Good Morning

$
```

This example shows a subsequent login on the same day by the same user. Note that no system messages are displayed.

```
$ LOGIN (RET)
Account or name: 301/370 (RET)
Password: (RET)

RSX-11M V4.2 BL38 Multiuser [3,54] System
25-MAY-85 19:36 Logged on Terminal TT45:

Good Evening

$ @LOGIN.COMD
$ SET TERMINAL/LOWER
$ ASSIGN RS: DB1:
$ @ <EOF>
$
```

This user has prepared a LOGIN.COMD file in his directory to set up the terminal and make a logical assignment. The LOGIN.COMD file is run automatically each time the user logs in. The commands shown are issued by the indirect command file; the @ <EOF> marks the end of the command file. See the Introduction to RSX-11M and the RSX-11M/M-PLUS Indirect Command Processor Manual for more information on indirect command files.

Notes

LOGIN and HELLO are identical. LOGIN does not work on systems without DCL, however.

LOGOUT counteracts LOGIN.

Once you have logged in and the system display has been printed on your terminal, the system searches the directory you logged in to for a file named LOGIN.COMD. This indirect command file can contain any commands you wish. This feature is commonly used to set up your terminal for regular use, as shown in the example. See the RSX-11M/M-PLUS Indirect Command Processor Manual for more information on indirect command files.

It is wise to log in occasionally without the slash to be sure you have not missed any important messages.

LOGIN (Cont.)

Error Messages

MCR -- Not logged in
DCL -- Not logged in

Explanation: The terminal is not logged in.

User Action: Log in using either LOGIN or HELLO.

LOG -- Account file open failure

Explanation: The system could not access the account file for some reason.

User Action: Try again. Wait and try again. If the message recurs, see your system manager.

LOG -- Invalid account

Explanation: The name or UIC, or the password, given in the command is not recorded in the account file.

User Action: Retype the command using correct data.

LOG -- Logins are disabled

Explanation: The system is being shut down, or a privileged user has issued SET NOLOGINS for some other reason.

User Action: You cannot log in. Try again later. Often, when logins are enabled again, the operator sends a message to all terminals.

LOG -- Other user logged on

Explanation: Another user is logged in. Only one user at a time can log in on a terminal.

User Action: Issue SHOW USERS to find out who is logged in. Issue SHOW TASKS ACTIVE to find out what tasks are active on the terminal. At your discretion, issue LOGOUT and then log yourself in.

LOG -- Terminal allocated to other user

Explanation: The terminal has been allocated (made private) by another user. You cannot log in on an allocated terminal.

User Action: Go to another terminal. Issue SHOW DEVICES TT: and find out which terminal has your terminal allocated.

LOGOUT

3.1.2 LOGOUT

LOGOUT logs the user out of a multiuser protection system and makes the terminal available to another user.

LOGOUT also aborts any active nonprivileged tasks running from the terminal, as well as dismounting any private volumes and deallocating any private devices allocated from the terminal (including those allocated by means of MOUNT/SHAREABLE).

Format

LO[GOUT] [/HOLD]

No prompts.

Command Qualifier

/[NO]HOLD

Used on remote or DECnet host terminals. If you include the qualifier, the terminal is logged out but the line is not disconnected. This means you can log back in without reconnecting the line. The default is /NOHOLD, meaning that when you log out, the line is also disconnected.

Examples

```
$ LOGOUT (RET)
Have a good afternoon
25-MAY-85 12:13 TT13: Logged off ZENITH
```

In this example, the user logged out of an RSX-11M system. The terminal is free for logging in. ZENITH is a DECnet node name. This information does not appear on systems without DECnet support.

```
$ LOGOUT (RET)
DMO -- TT10: Dismounted from DX0:
Have a good morning
25-MAY-85 11:17 TT10: Logged off ROMPER
```

In this example, the user had a volume mounted on device DX0:. The LOGOUT command dismounted the volume, as the message states. If the device was allocated, the LOGOUT command will deallocate it, but will not issue a message.

```
$ LOGOUT (RET)
13:29:36 Task "TT10 " Terminated
          Aborted via directive or CLI
$
Have a good afternoon
25-MAY-85 13:13 TT10: Logged off
```

In this example, the user had a task running from the terminal at the time the LOGOUT command was issued. The command caused the task to be aborted.

LOGOUT (Cont.)

Notes

LOGIN counteracts LOGOUT. You need type only LO to log out.

The LOGOUT command works only on terminals set to DCL. If the terminal is set to MCR, use the BYE command. Press CTRL/C to display the explicit monitor prompt thus enabling you to determine which CLI the terminal is set to.

3.1.3 ACNT, the Account File Maintenance Program

Account files on RSX-11M are created and maintained by ACNT, the Account File Maintenance Program. Whenever a user tries to log in, the system checks the user identification and the password against the account file to determine whether the user should be allowed access to the system, and with what privileges.

The account file describes all the UICs and associated passwords that have been authorized for a multiuser protection system. ACNT also associates a CLI (usually DCL or MCR) and a default device with each user.

Privileged users have full access to the functions of ACNT, as described in the RSX-11M/M-PLUS System Management Guide. Nonprivileged users must see their system manager to change any aspect of their account.

3.2 THE KEYBOARD

Most of the keys on the main keyboard are self-explanatory. They function much the same as they do on a typewriter. Only the most important differences from typewriter operation are noted here. See Table 3-1 for the names of keyboard characters as used in system documentation.

You should remember that a computer terminal is not a typewriter. A computer terminal has two functions, transmitting input to the computer and receiving output from the computer. Usually, when you press a key, the letter appears immediately on your terminal. When you pressed the key, you sent a character to some system task or program. One of the actions the system normally takes upon receiving a character is to send it back to your terminal as an echo. This usually happens so fast that it appears that you are typing on your terminal, but you are not. You are sending input to the system and it is sending output to your terminal. A terminal is an input/output device.

Occasionally, when heavy demands are being made on the system, there may be a noticeable lag between the input and the echo.

You should also remember that keys may behave differently when sending output to some task other than DCL. For instance, all the keypad keys, plus the ENTER, BACK SPACE and LINE FEED keys, have special meanings when you are running EDT, the standard DIGITAL editor. Special applications software at your installation may also use special meanings for certain keys. This chapter covers key use for DCL, also called monitor level.

TERMINAL OPERATIONS

3.2.1 RETURN and Command Line Length

The RETURN key has two functions. It is used in the conventional way to supply a line feed and a carriage return, but the key is also used to signal the system that you have finished typing a command. Commands can be quite varied in length; the RETURN key defines the length of the command. The symbol (RET) is used in this manual to signify that you press the RETURN key. Thus, the following command:

```
$ PRINT IZZY.TXT (RET)
```

is a complete command, and so is the following command:

```
$ PRINT/COPIES:2/FORM:1/DELETE IZZY.TXT (RET)
```

If you must enter a command that is longer than a single line on your terminal, you can override the effect of the RETURN key by ending the line with a hyphen before pressing the RETURN key. This works as follows:

```
$ PRINT/COPIES:2/FORM:1/DELETE IZZY.TXT, OZY.TXT, MADISON.AVE,-(RET)  
->/FUZZY.TXT;12, GUNGA.DIN;2 (RET)
```

The hyphen tells the system to postpone execution of the command until it receives a carriage return not preceded by a hyphen. In the example given, the user had more files to be printed together than could be specified on a single line. The hyphen called for the special continuation prompt (->) rather than execution of the command. The carriage return without the preceding hyphen then called for execution of the full command. For more information on command line continuation, see Chapter 1.

3.2.2 Line Terminators

You can terminate a command line with either the RETURN key, or the key variously labeled ESC, SEL, or ALTMODE. CTRL/Z and CTRL/C are also line terminators. Each of these terminators has different effects. The RETURN and ESC keys are discussed here. The control characters are discussed in Section 3.4.

The ENTER key on the keypad is identical with RETURN at DCL monitor level.

When you terminate a command line with the RETURN key, as soon as the command is processed, the implicit DCL prompt reappears, ready for another command. For example:

```
$ SHOW TIME (RET)  
09:59:21 13-JUN-85  
$
```

You need not wait for the return of the prompt to enter another command.

When you terminate a command line with the ESCAPE key, DCL does not prompt after the command has been processed. For example:

```
$ SHOW TIME (ESC)  
10:05:38 13-APR-85
```

TERMINAL OPERATIONS

The cursor or print head appears at the beginning of the line with the time on it.

The LINE FEED key provides a line feed without a carriage return. It cannot be used as part of the command line and is not a terminator.

3.2.3 DELETE Versus BACK SPACE

You should avoid the BACK SPACE key. This key is included on terminals for compatibility with other systems but is not used in DCL. The BACK SPACE key does not produce an echo, but it can have confusing results when used in a file or when entering a command. When you make a typing mistake, use the DELETE key to correct it, not BACK SPACE. (The BACK SPACE key may be used for special functions within tasks such as EDT.)

The DELETE key (labeled RUBOUT on some terminals) works in a slightly different fashion on video and hardcopy terminals.

On hardcopy terminals the DELETE key removes the last character typed and then verifies the removal by repeating the removed character on your terminal preceded by a backslash. This action is repeated each time you press the DELETE key. For instance, the following sequence:

```
$ RUN FEEBER   LE
```

appears on a hardcopy terminal as:

```
$ RUN FEEBER\RE\LE
```

But the operating system receives the command:

```
$ RUN FEEBLE
```

You can verify this by pressing a CTRL/R after the corrected line to retype it properly, without the intervening deleted characters. For example:

```
$ RUN FEEBER\RE\LE   
$ RUN FEEBLE
```

On a video terminal, the DELETE key actually erases the last character typed on your screen and you can replace it directly. The deleted characters are not repeated. For instance:

```
$ RUN FEEBER   LE
```

The sequence appears on a video screen in succession as follows:

```
$ RUN FEEBER  
$ RUN FEEBE  
$ RUN FEEB  
$ RUN FEEBL  
$ RUN FEEBLE
```

If you are using an editor that permits you to move the cursor around on the screen, the DELETE key deletes the first character immediately to the left of the cursor location.

TERMINAL OPERATIONS

3.2.4 CAPS LOCK and SHIFT

On most video terminals, the CAPS LOCK key causes all letters to be typed in uppercase. This key affects only letters, not numbers or symbols. You can engage CAPS LOCK and type 'RSX-llM' without disengaging it, even though the numbers and the hyphens are lowercase.

The SHIFT keys set your terminal to uppercase for the keys not affected by CAPS LOCK, as well as for individual alphabetic characters.

On most hardcopy terminals, CAPS LOCK sets all keys to uppercase.

Different terminal models vary widely in their treatment of CAPS LOCK or the equivalent key. You should experiment on your terminal.

3.2.5 REPEAT

The REPEAT key on the VT52 and LA36 repeats the action of any other key. You can use it to insert a series of blanks, characters, carriage returns, or whatever. Use it by holding down REPEAT and the key whose action you wish to repeat. It is particularly handy with the DELETE key. Normally, a DELETE-REPEAT combination deletes back to the left margin of your terminal and no further.

Most keys on a VT100- or VT200-series terminal repeat automatically if you keep the key depressed. However, if the terminal is set to noautorepeat mode, pressing a key will move the cursor one space only. There is no REPEAT key on either a VT100- or VT200-series terminal.

3.2.6 SCROLL and NO SCROLL

The SCROLL key on VT52s is used together with the SET TERMINAL/HOLD_SCREEN command. This command limits the terminal to one screenful of information at a time, allowing you to read output that requires more than one screen's worth of space. If HOLD_SCREEN is set, the SCROLL key moves the display up one line at a time. SHIFT and SCROLL together bring up a new screenful of information. (A CTRL/C or SET TERMINAL/NOHOLD_SCREEN returns the terminal to normal.)

The NO SCROLL key on the VT100 and the HOLD SCREEN key on the VT200-series terminal simply causes the terminal to stop scrolling. When you are ready for the terminal to scroll, press the NO SCROLL key or the HOLD SCREEN key again.

See also the discussion of the CTRL/S and CTRL/Q keys in Section 3.4.2.

See the documentation provided with your terminal for information on controlling scrolling on other terminal models.

3.2.7 Other Keys

The COPY key is used only on terminals with a copier.

Other keys are listed in Table 3-1, along with common names for them.

TERMINAL OPERATIONS

Table 3-1
Keyboard and Keypad Characters

Character	Name	Comments
^	Circumflex	<p>On some keyboards, this character appears as an up-arrow. It should not be confused with the cursor-control keys on the keypad, which also have arrows.</p> <p>At monitor level, the CTRL key sometimes echoes as a circumflex when struck in combination with a letter.</p> <p>If you wish to include CTRL/a characters in a batch job, the letter preceded by the circumflex will be treated as a CTRL/a.</p> <p>In BASIC programs, the circumflex is the mathematical operator for exponents.</p>
\$	Dollar sign	Standard DCL prompt. Established in STARTUP.CMD. It may be different on your system.
>	Right-angle bracket	General term.
	Greater-than	ASCII name of " character. Used in mathematical and logical expressions.
	Implicit monitor Prompt	At monitor level, signifies that system is ready to accept commands. The explicit monitor prompts are MCR> and DCL>.
<	Left-angle bracket	General term.
	Less-than	ASCII name of character. Used in mathematical and logical expressions.
.	Period	In text.
	Decimal point	In numbers.
	Dot	In file specifications, terminator between file name and file type.

(continued on next page)

TERMINAL OPERATIONS

Table 3-1 (Cont.)
Keyboard and Keypad Characters

Character	Name	Comments
	Dot (Cont.)	The period keys on the keyboard and keypad are identical at monitor level.
@	At sign	Indirect command file operator.
	Commercial at	ASCII name. Not in the international character set.
#	Number sign	Not pound sign.
&	Ampersand	General term.
%	Percent sign	General term.
	Wildcard	Stands for "match one character in this position" in file names and file types.
*	Asterisk	General term.
	Wildcard	Stands for "match any or no characters in this position" in file names and file types, or for any group, member, or version number.
_	Underscore	Used to tie command words together in DCL.
-	Hyphen	DCL continuation character. Used for command lines too long for terminal line. Overrides RETURN. Command not entered until monitor receives RETURN without preceding hyphen.
-	Minus	Mathematical operator for subtraction.
+	Plus	Mathematical operator for addition.
:	Colon	Marks beginning of argument in DCL command.
=	Equal sign	Mathematical and logical operator. Also used extensively in MCR syntax.

(continued on next page)

TERMINAL OPERATIONS

Table 3-1 (Cont.)
Keyboard and Keypad Characters

Character	Name	Comments
{ }	Braces	General term.
()	Parentheses	General term. Also delimiters for a series of arguments or parameters attached to DCL commands or qualifiers.
[]	Square brackets	Delimiters for UIC and UFD numbers.
/	Slash	Marks beginning of qualifier in DCL commands. Must be preceded by space in MCR commands.
\	Backslash	On hardcopy terminals, delimiter for deleted characters.

3.3 THE KEYPAD

Some terminals have a second set of keys to the right of the main keyboard, arranged much like a calculator's keypad. These keys make up the keypad.

The keys of the alternate keypad are available for special functions. Some functions of the operating system use these keys. They may be used in many other ways, depending on the particular installation. The following description covers only the uses of the keypad by functions at the monitor level.

The following list describes only the uses of the alternate keypad while you are in DCL.

Keypad Key	Function
NUMBERS	Work in the same way as the numbers on the regular keyboard.
PERIOD	Works in the same way as the period on the regular keyboard. It is used for including the decimal point in numbers.
ENTER	Works the same as the RETURN key on the main keyboard.
PF KEYS	Reserved for special functions. They are not used at monitor level.
ARROW KEYS	Used for cursor control by EDT, but are otherwise available for special functions.

The VT200-series terminals have both the alternate keypad and, located between it and the keyboard, another keypad with editing functions printed on the keys. These keys make up the editing keypad.

TERMINAL OPERATIONS

The six labeled keys on the editing keypad are available for special functions. They are defined by EDT as well as by many other applications.

The three blank keys (blue, red, and grey) on the VT52 are reserved for special functions.

The keys of both the alternate and editing keypads can have confusing effects if pressed accidentally. These keys each send an ESCAPE character and one or more characters to the operating system. The ESCAPE is a nonprinting character, but associated characters echo on the terminal and may make the operating system behave as if you had typed those letters.

Here is how these special keys are set up on the VT52, VT100-, and VT200-series terminals:

Key	Effect
up-arrow	types [A
down-arrow	types [B
right-arrow	types [C
left-arrow	types [D
blue blank	types P
red blank	types Q
grey blank	types R
PF1	types OP
PF2	types OQ
PF3	types OR
PF4	types OS
FIND	types [1~
INSERT	types [2~
REMOVE	types [3~
SELECT	types [4~
PREV	types [5~
NEXT	types [6~

3.4 THE CTRL KEY AND CONTROL CHARACTERS

The key marked CTRL is called the control key. It is used much like the shift key on a regular typewriter. If you hold down the CTRL key and then press another key, you are sending a command to the operating system. The CTRL key alone has no effect.

The combination is designated by CTRL/a, where a is the chosen letter key.

When the combination is echoed on your terminal, the CTRL key is represented by the circumflex (^). Press a CTRL/U and you see a ^U on your terminal.

The operating system recognizes 11 letters combined with the CTRL key. The letters are C, I, K, L, O, Q, R, S, U, X, and Z. These are discussed below in approximate order of importance.

With the exception of CTRL/Z, the control characters are commands directed to the operating system and not to any task you may be running from your terminal. This means that they usually work without interfering with such tasks. For instance, CTRL/O is used to skip over terminal output. It skips over any output, whether sent by DCL, the Task Builder, an editor, or any other system function. It only skips output. It does nothing else.

TERMINAL OPERATIONS

NOTE

Although you must press two keys to enter a control character, the system considers a control character as a single character.

3.4.1 CTRL/C

CTRL/C is a line terminator. Its main use is to gain the attention of the operating system.

In most cases, CTRL/C gives you access to the operating system without interrupting any tasks you may have running from your terminal.

Pressing a CTRL/C causes the CLI to prompt for command input with an explicit prompt. This prompt will be either DCL> or MCR>, depending on which CLI your terminal is set to. You can then type a command without interrupting any other tasks active at your terminal.

This is particularly convenient when you want to abort a task running from your terminal.

CTRL/C also assures you that the command you enter will go to DCL when the DCL> prompt is showing.

If a terminal is receiving output from a task, CTRL/C interrupts that output. After DCL processes the input directed to it, the system resumes the interrupted output. No output is lost.

CTRL/C has special effects in two cases:

1. If you have placed your VT52 in hold-screen mode with the command SET TERMINAL/HOLD_SCREEN, a CTRL/C and a RETURN takes the terminal out of hold-screen mode. It is easier to press a CTRL/C than to type SET TERMINAL/NOHOLD_SCREEN. It does not matter whether you type in a command in response to the DCL> prompt; hold-screen mode is disabled anyway. Hold-screen mode is available in this form on VT52s only.
2. If a CTRL/S has temporarily halted the output of data to the terminal, a CTRL/C starts it up again. In this case, CTRL/C is the same as CTRL/Q. See the discussion of CTRL/S and CTRL/Q in the next section.

3.4.2 CTRL/S and CTRL/Q

This pair of control characters is used to alternately stop and start output to a terminal. CTRL/S delays the output and CTRL/Q resumes it. It is particularly useful on video terminals where the output may go by too fast to be read, but it works on hardcopy terminals as well.

The CTRL/S and CTRL/Q keys function like the NO SCROLL key on the VT100-series terminals and the HOLD SCREEN key on the VT200-series terminals.

If CTRL/S has temporarily halted the output of the data to your terminal, CTRL/C has the same effect as CTRL/Q; pressing CTRL/C also resumes the flow of information.

TERMINAL OPERATIONS

If you use these control characters, you do not skip over any of the output. All output that should appear, will appear. It is simply held back until you are ready to look at it. (For skipping output, see the discussion of CTRL/O in the next section.)

If your terminal appears to be doing nothing, you may have inadvertently pressed CTRL/S. Press CTRL/Q to resume output to your terminal.

Note that your terminal must be set /TTSYNC for CTRL/S and CTRL/Q to work. Because /TTSYNC is the default, you do not need to set this characteristic unless you have changed it previously with /NOTTSYNC.

3.4.3 CTRL/O

This control character is used to skip over unwanted output to your terminal. It is analogous to the fast-forward control on a tape recorder.

If you type a single CTRL/O in response to undesired output, the output stops appearing on your terminal, but the system continues to generate the output. It can do this much faster when it does not have to print it on the terminal. If you type another CTRL/O, the output picks up at the point the operating system has reached, not the point at which you typed the last CTRL/O. If you do not type another CTRL/O, the system continues to generate the output until it reaches the end, whereupon a prompt appears and you may continue in response to the prompt.

If you are printing multiple files on your terminal, the CTRL/O will be cancelled at the end of each file. For example, if you have typed the following:

```
$ TYPE *.LST (RET)
```

the system will start with the first file of the type .LST in your directory. If you suppress this output with a CTRL/O, the system will start printing again as soon as it reaches the beginning of the next file of the type .LST.

3.4.4 CTRL/Z

CTRL/Z has several functions when used in the following instances:

- Pressing CTRL/Z indicates end-of-file. It is used to indicate to system tasks, such as the MACRO-11 Assembler or the Task Builder, that you have finished and that the task should exit.
- A CTRL/Z in response to a prompt or at the end of a command line cancels the command.
- CTRL/Z is also used as a command within many system tasks, particularly the editors.
- If your terminal appears to be hanging, a CTRL/Z will often clear it.

TERMINAL OPERATIONS

- A CTRL/Z in response to a request for input from an indirect command file cancels the execution of the remainder of the file.
- Fundamentally, CTRL/Z signals the system that you have finished typing input.

3.4.5 CTRL/U

CTRL/U deletes the current line on your terminal. It is as if the line had never been typed. The system responds with a carriage return and a line feed so that the line can be retyped.

If you use CTRL/U with a CREATE command when typing in a sequential file, the line you have deleted will appear as a blank line within the file.

In three instances, a ^U may appear on your terminal without your having typed in a CTRL/U.

1. A system-generation option automatically sends a CTRL/U to any terminal that has been awaiting input past a time-out limit. This limit can be set by your system manager in seconds to anything from one second to 21 minutes. If this option is included, any line that is not terminated within the time-out limit is wiped out, as indicated by the CTRL/U that appears at the terminal. If you have typed in a line and then deleted it with the DELETE key, the system still interprets this as a line awaiting input and sends the CTRL/U when the time-out limit passes.
2. If you try to type more than 80 characters on a line without using a hyphen to continue the line, the system automatically cancels the command and indicates this by sending a ^U to the terminal. See Section 1.2.13 for more information on line continuation.
3. Finally, if there is no available pool space when you type in a command, the system sends a ^U to your terminal informing you that your command has been cancelled. In this case, the condition will usually be corrected shortly. Try typing the command again. If this does not work after some reasonable interval, inform your system operator of the problem.

3.4.6 CTRL/R

Typing CTRL/R before ending a line causes the line to be retyped as it currently stands, that is, with any corrections you have made in it.

This command is most useful on hardcopy terminals, where the DELETE key causes characters to be retyped as they are deleted. You can also use it on a video terminal to confirm that a line is what you think it is.

For example:

```
$ THINK\KM\NK CTRL/R
$ THINK
```

You may then continue typing the line.

TERMINAL OPERATIONS

Although you see two copies of the line, there is only one line as far as the system is concerned.

If you are typing a command on your terminal when a broadcast or system message arrives, you will see a ^R and a second copy of your command line. Simply continue typing the command.

3.4.7 CTRL/X

A CTRL/X clears your terminal's typeahead buffer. This buffer is used to store input characters temporarily until your terminal can pass them to the operating system.

Occasionally, this buffer gets full. When your typeahead buffer is full, you cannot type on your terminal and any key you strike rings the bell (buzzer, beeper). When this happens, type a CTRL/X and you may continue using your terminal.

See Section 3.8.1, the description of the SET TERMINAL command, for more information on the typeahead button.

3.4.8 CTRL/L

A CTRL/L performs a form feed. Form feeds are interpreted differently by different kinds of output devices. A form feed will always add several line feeds at the point at which it is inserted. On a line printer, a CTRL/L will usually move the paper past the next perforation.

3.4.9 Other Control Characters

CTRL/G inserts a bell (buzz, beep) in a file or message. CTRL/G cannot be the first character in a line.

CTRL/I is the same as TAB. The cursor or print head moves to the next tab position. Tabs are set every eight places.

CTRL/K inserts a vertical tab by performing four line feeds.

CTRL/M is the same as a RETURN.

HELP

3.5 HELP

HELP displays information about DCL and MCR commands and may display other information about the operating system.

HELP is the only command that you can issue without logging in.

Format

```
$ HELP[/OUTPUT:filespec][qualifier] [%] [parameter1] [...parameter9]
```

Command Qualifiers

```
/OUTPUT:filespec
/LOCAL
/GROUP
/CLI:cliname
/MCR
/DCL
/FILE:filespec
/filename
```

Parameters

No parameters are required for the HELP command and there are no prompts. You can get help on any given DCL command or subtopic, however, by typing a question mark (?) in response to the prompts from DCL. You can also get help by typing a question mark in response to the system prompt (>, if MCR or \$, if DCL). Examples follow at the end of this section.

The HELP command without qualifiers or parameters displays the list of commands for the command line interpreter (CLI) for which your terminal is set, DCL, MCR, or some alternative CLI in use at your installation. This display contains information on getting more help.

The displays for all forms of the HELP command are stored in help files, which are text files in help file format. See Section 3.5.1 for information on writing help files.

parameter1

If you include a parameter in the HELP command, you will jump over the initial HELP display to a display concerning the parameter you have included. Examples follow at the end of this section.

Usually this parameter will be a command name. When seeking help on a command, you should always start with a single parameter. The display always includes directions for getting any available further help.

parameter9

There can be up to nine levels of help. In any case, the number of parameters you enter determines the display shown; you always jump over intervening displays. See Section 3.5.1.

HELP (Cont.)

Command Qualifiers

/OUTPUT:filespec

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

The following qualifiers are mutually exclusive. These qualifiers have the effect of naming a file where the requested help text is located. No matter which qualifier you use, you can still include up to nine parameters.

If you are in DCL and use none of these qualifiers, the default command is HELP/DCL. If you are in MCR and use none of these qualifiers, the default command is HELP/MCR.

/LOCAL

%

Specifies that the help text is in the file HELP.HLP in the default directory on the default volume. You can specify a local help file in two ways: HELP/LOCAL or HELP %. Examples follow at the end of this section.

/GROUP

Specifies that the help text is in the file HELP.HLP in the directory [current group,1] on the default volume. All users with the same group number have access to this file by typing HELP/GROUP. Examples follow at the end of this section.

/CLI:cliname

Specifies that the help text begins in the file LB:[1,2]cli.HLP. This qualifier is for installations with alternate CLIs for which HELP is provided. The cliname can be from 1 through 6 Radix-50 characters. Examples follow at the end of this section.

/MCR

Specifies that you wish help on an MCR command. For this qualifier, the help text begins in the file LB:[1,2]MCR.HLP. This is the default for terminals set to MCR.

/DCL

Specifies that you wish help on a DCL command. For this qualifier, the help text begins in the file LB:[1,2]DCL.HLP. This is the default for terminals set to DCL.

/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. When using this qualifier, you cannot specify the device, UFD, or extension, only the filename.

HELP (Cont.)

Examples

```
$ HELP (RET)
```

Help is available for the following DCL commands:

ABORT	ALLOCATE	ANALYZE	APPEND	ASSIGN
BACKUP	BASIC	BROADCAST	CANCEL	COBOL
CONTINUE	CONVERT	COPY	CREATE	DEALLOCATE
DEASSIGN	DEBUG	DELETE	DIFFERENCES	DIRECTORY
DISMOUNT	EDIT	FIX	FORTRAN	HELP
HOLD	INITIALIZE	INSTALL	LIBRARY	LINK
LOGIN	LOGOUT	MACRO	MCR	MOUNT
PRINT	PURGE	RELEASE	REMOVE	RENAME
REQUEST	RUN	SET	SHOW	SORT
START	STOP	TYPE	UNFIX	UNLOCK

For information on a command, type HELP commandname. Additional help on a particular qualifier is often available by typing

HELP commandname qualifier.

You can also get any DCL help on DCL commands by typing a ? in response to a prompt.

For information on utilities, system tasks, and other system information, type HELP MORE. For the short forms of some commands, type HELP BRIEF. For help on logging in, type HELP LOGIN.

This example shows the initial display from HELP. The user typed HELP without parameters or qualifiers at a DCL terminal.

```
$ TYPE (RET)
File(s)? ? (RET)
```

```
TYPE[/qualifier[s] filespec[s]
  /DATE:dd-mmm-yy
  /SINCE:dd-mmm-yy
  /THROUGH:dd-mmm-yy
  /SINCE:dd-mm-yy/THROUGH:dd-mmm-yy
  /TODAY
  /EXCLUDE:filespec
  /NOWARNINGS
  /SHARED
```

The TYPE command displays the contents of text files on your terminal.

```
File(s)? (RET)
```

This example shows the display that results from typing in a question mark (?) in response to a prompt from DCL. The same display results from typing HELP TYPE.

HELP (Cont.)

```
$ SHOW (RET)
Function? ? (RET)
SHOW thing
```

The SHOW command can be used to show something. The following things can be shown with this command:

ASSIGNMENTS	DEVICES	PARTITIONS	TASKS
CLOCK QUEUE	GROUPFLAGS	PROCESSOR	TERMINAL
[DAY]TIME	HOST	QUEUE	USERS
DEFAULT	MEMORY	SYSTEM	

```
Abbreviation: S
Function? ?TIME (RET)
```

```
SHOW DAYTIME
SHOW TIME
```

The SHOW TIME command displays the current time and date. The time is in 24-hour format and the date is formatted as dd-mmm-yy.

```
Function? (RET)
```

This example shows the display that results from typing a question mark (?) and a keyword in response to a prompt from the SHOW command. The same display results from typing HELP SHOW TIME.

3.5.1 Help File Format: Writing a Help File

The HELP facility requires a specially formatted file or files called help files. A help file is a text file with up to nine levels of nesting. The format is much the same as a conventional outline. The file starts with a main topic followed by any subtopics. Each subtopic can include sub-subtopics, and so on through all nine levels.

The format is the same for any help file: local, group, cli, or special purpose.

You can put all help text in a single file, but for faster processing on larger help files, you will probably want to use indirect referencing. This is explained further below. The first (or only) help file accessed is called the root file.

In addition to the help text, a help file includes the numbered level indicators, as well as other operators to indicate synonyms and indirect references. The operators must appear in the first column. Anything else that appears in the first column is ignored by HELP. The help operators include the level numbers from 1 through 9, the number sign (#) indicating a synonym, and the at sign (@) indicating an indirect file reference.

The level number in the first column must be followed by at least one blank or tab before the parameter. The parameter must be all uppercase characters without intervening blanks or tabs. Any text that appears before the first level-number line is printed in response to a HELP command without any parameters.

TERMINAL OPERATIONS

Parameters at each level must have different names, but names can be duplicated at each different level, as shown in the example. You need type only the distinguishing portion of a parameter.

HELP interprets its command line in a strictly nested fashion. When interpreting parameters, HELP searches down through the help file for the first line starting with the numeral 1 followed by the specified parameter. If such a line exists, HELP displays all the text between that line and the next level number. The lines with level numbers and parameters in them are never displayed.

If you specify more than one parameter, HELP searches for a line starting with the numeral 1 followed by the first parameter and then continues the search for a line starting with the numeral 2 followed by the second parameter. If HELP encounters a line starting with the numeral 1 before encountering the appropriate level-2 line, the search stops. If the line starting with a 2 followed by the parameter given is encountered, HELP displays the text between that line and the next level-number line. This same procedure is followed for all nine parameters.

The level-number line is not always followed by text. The following line may begin with a number sign (#) or an at sign (@). If the line begins with a #, HELP substitutes the synonym given there for the original parameter and continues the search. If the following line begins with an @, HELP then branches to the file specified by the @ and continues its search in that file.

All these operators are included, with further commentary, in the following help file example:

```
Here is the text displayed by typing HELP without parameters.
Introductory information on the further use of the help file
usually appears here. For better appearance, the first
character in each line is usually a tab or blank, but there is
no technical requirement that you do this. The HELP processor
ignores any first character except the numerals 1 through 9
and the symbols @ and #. It is good practice to limit text to
about 20 lines so that the display text can be displayed on a
video terminal without scrolling and with room left to enter a
command without losing help text. Text at each level should
refer to all help available at the next level.
```

Further help is available. Type

```
HELP FIRST-LEVEL1      HELP SECOND-LEVEL1
HELP INDIRECT          HELP SYNONYMS
HELP PURPLE            HELP MAUVE
```

1 FIRST-LEVEL1

```
This is the first level-1 help text. The first line of this
text must begin with the numeral 1. This is the text
displayed if you type
```

```
$ HELP FIRST-LEVEL1
```

```
The parameter must be all uppercase characters (CAPITALS). It
is not necessary to type the full text of the parameter to
display the text, but you must type enough of it to
distinguish the parameter from all other parameters at the
same level. In this example, typing
```

```
$ HELP FIRST
```

TERMINAL OPERATIONS

would be enough. The parameter must be separated from the level number by one or more blanks or tabs. The parameter itself cannot include any blanks or tabs, but phrases can be tied together with hyphens or underscore characters. If you do this, make it clear in your reference to that level.

Further help is available. Type

HELP FIRST-LEVEL1 FIRST-LEVEL2

2 FIRST-LEVEL2

This is the text displayed if you type HELP FIRST FIRST. This is the first level-2 help text.

1 SECOND-LEVEL1

This is the text displayed if you type HELP SECOND. This is the second level-1 help text.

Further help is available. Type

HELP SECOND-LEVEL1 SECOND-LEVEL2

2 SECOND-LEVEL2

This is the text displayed if you type HELP SECOND SECOND. This is the second level-2 help text.

1 INDIRECT

Help files can also include indirect references to other help text. These indirect references greatly decrease search time because once the HELP processor sees the indirect reference, the calling file (this file) is left behind. The filespec can be any legal filespec. The default file type is .HLP. If you use a file type other than .HLP, that type becomes the default file type for that session of HELP.

The file referred to in this manner has the same format as other help files, except that if the indirect file contains level numbers, these must be greater than the level number in the calling file. For instance, if the calling file contains the lines

```
1 INSTANCE
@INSTANCE
```

then INSTANCE.HLP can contain only level numbers 2 through 9.

Further help is available. See

HELP INDIRECT EXAMPLE

2 EXAMPLE

```
@EXAMPLE.HLP
```

1 SYNONYMS

HELP also permits cross-references between two or more similar or synonymous parameters. The format for synonyms appears below. You can greatly assist the user by trying to imagine what might be typed in seeking particular help text and including the most likely possibilities as synonyms.

The synonyms must all precede the main entry to which they refer. The main entry and all synonym entries must have the same level number.

TERMINAL OPERATIONS

1 MAUVE
#PURPLE

·
·
·

1 MAGENTA
#PURPLE

·
·
·

1 PURPLE

Color of a hue between blue and red. Shades of purple include mauve, violet, plum, lilac, puce, amethyst, and magenta.

BROADCAST

3.6 BROADCAST

BROADCAST displays a specified message at one or more terminals.

Format

```
BROADCAST
To? [ttnn:]
Message? message

BROADCAST/qualifier
Message? message

BROADCAST @filespec
```

Command Qualifiers

```
/ALL
/LOGGED_IN
```

Parameters

message

The message must fit on a single line, but the final character can go in column 80. The message can include any printing character. Lowercase letters are converted to uppercase.

If you do not specify a destination for the message, the message is sent to pseudo device CO:, the operator's console. This is the equivalent of the REQUEST command. See Section 3.7.

ttnn:

Specifies the terminal to which you wish the message to be sent. Terminals need not be logged in to receive messages, but they must be turned on (powered).

@filespec

Specifies the name of an indirect command file. All users can send multiple messages or multiple copies of the same message using this method. The file should contain messages in the following format:

```
TTnn:message
```

Privileged users can also use the privileged qualifiers in the following formats:

```
ALL:message
LOGGED_IN:message
```

The indirect command file cannot include any Indirect directives or labels, only destinations and messages. The destination cannot be preceded by tabs or blanks.

BROADCAST (Cont.)

Command Qualifiers

/ALL

This privileged qualifier sends the same message to all powered terminals, excluding slaved terminals.

/LOGGED_IN

This privileged qualifier sends the same message to all logged-in terminals.

Examples

```
$ BROADCAST (RET)
To? TT20: (RET)
Message? Meet me in the alley behind the barn. (RET)
```

The message is printed on TT20:

```
30-OCT-85          FROM: TT10:          TO: TT20:
MEET ME IN THE ALLEY BEHIND THE BARN.
```

\$

As the message arrives, the terminal's audio signal sounds. Notice that the message is printed in uppercase characters on the receiving terminal. If you enclose the message in quotation marks ("), it is printed exactly as you entered it, including the quotation marks.

```
$ BROADCAST TT12: THIS IS A ONE-LINER. (RET)
```

This is the one-line form of the command.

```
$ BROADCAST/LOGGED_IN (RET)
Message? Everybody take the rest of the day off. (RET)
```

The message is printed on all logged-in terminals. This is a privileged command.

```
$ BRO @DAILY (RET)
```

The file DAILY.CMD contains the following messages:

```
TT11:WHERE IS THE DITKO CONTRACT?
TT13:MEET ME FOR LUNCH AT 11:30.
TT27:THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
TT35:GET BACK TO WORK.
TT41:I LOST MY GREEN EYESHADE. HAVE YOU SEEN IT?
```

The messages are sent to the designated terminals. Privileged users can also include messages preceded by ALL: and LOGGED_IN:.

BROADCAST (Cont.)

Notes

B is the short form of BROADCAST.

Only terminals can receive messages.

If you wish your message to include lowercase characters when printed, enclose the message in quotation marks (").

If the message cannot be broadcast within 10 seconds, the system displays the following message at the initiating terminal:

BRO -- Terminal is busy --TTnn:

If a user specifies multiple destinations, the system returns an error message for each busy terminal.

The BROADCAST command uses the write-breakthrough feature of the terminal driver. This means the message breaks through any kind of I/O at the destination terminal. If you are editing, for instance, the message may appear in the middle of your text, but in fact it has no effect on the text you are editing. You can issue a SET TERMINAL/NOBROADCAST command if you do not wish to receive broadcasts. See Section 3.8.1.

This command invokes the MCR BRO task.

Error Messages

BRO -- Command input error

Explanation: The BROADCAST task did not receive the command line.

User Action: Often this message results from a missing indirect command file. Locate the file; or, check for proper syntax and enter the command again.

BRO -- Command syntax error

Explanation: The command syntax was not correct.

User Action: Check for proper syntax and enter the command again.

BRO -- Illegal device specified

Explanation: The destination device was not a terminal.

User Action: Check for proper device and enter the command again.

BRO -- Privileged command

Explanation: Nonprivileged users cannot send messages to all connected or logged-in terminals.

User Action: Use an indirect command file to send messages to multiple terminals from a nonprivileged terminal.

BROADCAST (Cont.)

BRO -- User not receiving messages

BRO -- Ttnn:message

Explanation: The message was sent to a terminal set to NOBROADCAST.

User Action: Wait and try again.

REQUEST

3.7 REQUEST

REQUEST sends a message to pseudo device CO:, the operator's terminal.

Format

```
REQUEST
Message? message
```

```
REQUEST message
```

Parameter

message

The message must fit on a single line, but the final character can go in column 80. The message can include any printing character. Lowercase letters are converted to uppercase.

The message is time-stamped and indented on the operator's terminal.

Example

```
$ REQUEST Charlie, the tape drives are down again. (RET)
```

The message appears on the operator's console in the following format:

```

      .
      .
      .
13:30:45      Login user ISHMAEL
13:32:45      Login user MOBYDICK
13:32:47

      19-APR-85          FROM: TT56:          TO: CO:
      CHARLIE, THE TAPE DRIVES ARE DOWN AGAIN

13:36:22      Login user QUEEQUEG
14:36:42      Login user PEQUOD
      .
      .
      .
```

The user at TT56: observed a problem on the system and reported it to the operator with REQUEST. The message appears on the operator's console, time-stamped and indented so that it stands out from other messages.

REQUEST (Cont.)

Notes

The REQUEST command is very similar to the BROADCAST command. The main difference is that no destination is required. REQUEST works best on systems when the operator's console is a hardcopy terminal.

CO: is usually, but not always, TT0:. In most installations, however, any REQUEST message will come to the attention of the operator or system manager.

See the discussion of COT, the Console Output Task, in the RSX-11M/M-PLUS System Management Guide for more information on CO:.

3.8 SET AND SHOW TERMINAL

SET TERMINAL and SHOW TERMINAL are complementary commands. SET TERMINAL sets terminal characteristics and SHOW TERMINAL displays information about terminal characteristics. For detailed information about these characteristics, see the discussion of the full-duplex terminal driver in the RSX-11M/M-PLUS I/O Drivers Reference Manual.

SET TERMINAL

3.8.1 SET TERMINAL

SET TERMINAL sets various attributes of your terminal. Privileged users can set attributes for any terminal.

Format

```
SET
Function?  TERMINAL[:TTnn:]
Terminal Attribute?  attribute[/attribute[s]]

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

Attributes

Group 1: Common Use

```
/[NO]BROADCAST          /CLI:cliname
/DCL                    /[NO]HOLD_SCREEN
/INQUIRE               /[NO]LOWER_CASE
/MCR                   /[NO]PRIVILEGED
/[NO]UPPERCASE         /WIDTH:n
/SPEED:(transmit,receive)
```

Group 2: Terminal Setup

```
/ASR33                 /ASR35
/KSR33

/LA12                  /LA30P
/LA30S                 /LA34
/LA36                  /LA38
/LA50                  /LA100
/LA120                 /LA180S
/LA210

/LN03

/LQP02                 /LQP03

/PRO_SERIES

/VT05B                /VT50
/VT52                 /VT55
/VT61                 /VT100
/VT101                /VT102
/VT105                /VT125
/VT131                /VT132
/VT200_SERIES

/ADVANCED_VIDEO       /ANSI CRT
/[NO]AUTOBAUD         /BLOCKMODE
/CRFILL:n             /DEC CRT
/EDIT MODE            /[NO]FORM FEED
/[NO]HARDCOPY         /HOSTSYNC
/LFFILL               /MODEL:arg
/PAGE LENGTH:n       /PRINTER PORT
/REGIS                /[NO]SCOPE
/SOFT_CHARACTERS     /[NO]TAB
/TTSYNC
```

SET TERMINAL (Cont.)

Group 3: Task Setup

/[NO]ECHO	/[NO]EIGHT_BIT
/[NO]ESCAPE	/[NO]FULL_DUPLEX
/[NO]INTERACTIVE	/[NO]LOCAL
/PARITY:arg	/[NO]PASSALL
ODD	/[NO]PASTHRU
EVEN	/[NO]REMOTE
/[NO]SLAVE	/[NO]TYPEAHEAD
/[NO]WRAP	

Command Option

TTnn:

Specifies the terminal for which an attribute is to be set.

For nonprivileged users, only your own terminal number or TI: is permitted here. Since TI: is the default, nonprivileged users need not give a terminal number. Privileged users can name any terminal.

Attributes

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

Group 1: Common Use

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

/[NO]BROADCAST

Establishes whether you wish to receive broadcast messages on your terminal. The default is to receive broadcasts.

This attribute restricts messages from BROADCAST and SHUTUP only. All BROADCAST messages are intercepted when you set your terminal /NOBROADCAST. Senders are notified that you are not receiving messages. See the examples. As for SHUTUP, all messages about the pending shutdown of the system are intercepted except those issued in the last five minutes of SHUTUP. These are the only messages intercepted when you set your terminal /NOBROADCAST. Messages from other tasks are not affected.

/CLI:cliname

Sets a terminal to an alternate command line interpreter (CLI). The cliname is from one to six Radix-50 characters. See your system manager for information on any alternative CLIs at your installation.

/DCL

Sets terminal to DCL. Since this is a DCL command, it is mainly useful for privileged users who wish to set some other terminal to DCL, but it is not a privileged command.

SET TERMINAL (Cont.)

/[NO]HOLD_SCREEN

Sets a VT50, VT52, or VT55 terminal to HOLD_SCREEN.

Video terminals can present output much faster than you can read it. HOLD_SCREEN lets you accept output on your VT50, VT52, or VT55 terminal one line or one screenful at a time.

/NOHOLD_SCREEN is the default. Output appears at the bottom of the screen and scrolls off the top at the rate it is sent to the terminal.

/HOLD_SCREEN first clears the screen, then fills in from the top of the screen to the bottom and stops. If you want to see another line, press the SCROLL key once. If you want to see another screenful, press the SCROLL key while holding down the SHIFT key. Data is being held back. It is not being lost.

If you want to disable HOLD_SCREEN, either issue SET TERMINAL/NOHOLD_SCREEN or press CTRL/C.

You can also control output with the CTRL/S and CTRL/Q keys. On terminals in the VT100-series or the VT200-series, you can use the NO_SCROLL or the HOLD_SCREEN keys, respectively. See Section 3.2.6 for more information.

/INQUIRE

Sends a query to the terminal to find out what model it is and then issues the SET TERMINAL command for that terminal model. The following DIGITAL terminals are set explicitly: VT52, VT61, the VT100-series (VT100, VT101, and VT105 are all set VT100), the VT200-series, the DECmate II and the Rainbow 100-series (these last two are set VT102), the Professional 300-series, LA34, LA38, LA100, and LA120. All other terminals are set /HARDCOPY, /WIDTH:80, and are considered to be "unknown" models.

The /INQUIRE attribute is perhaps the most useful of all. If you issue the command SET TERMINAL/INQUIRE, the operating system sets all appropriate attributes for that type of terminal. These attributes include all attributes listed under **Terminal Setup**. In addition, SET TERMINAL/INQUIRE sets the terminal width at 80 columns. It is good practice to include this command in your LOGIN.CMD file.

/LOWERCASE /NOUPPERCASE

Characters typed on the terminal in lowercase appear in lowercase. It is common practice to set terminals to lowercase in a LOGIN.CMD file. Most system tasks will accept input in lowercase. Note that some terminal escape sequences use lowercase characters. If, for instance, the keypad commands in EDT do not seem to be working, you may have to set the /LOWERCASE attribute.

SET TERMINAL (Cont.)

/MCR

Sets terminal to MCR. Terminal no longer accepts DCL commands. The counteracting MCR command is SET /DCL=TTnn:.

/[NO]PRIVILEGED

Sets a terminal privileged or nonprivileged. A privileged terminal can issue all DCL and MCR commands. The default is determined by whether you log in privileged or nonprivileged. Privileged users are those whose UIC has a group number equal to or lower than 10. (The group number is the first number of the UIC.) Setting this attribute is a privileged option.

/SPEED:(transmit,receive)

Sets terminal's rates for transmission and reception of data. This software speed setting must be matched by the hardware speed setting, which is controlled by switches on the terminal itself. The first number in parentheses is the transmission rate, the second is the reception rate. Note the use of the comma. When you issue SHOW TERMINAL/SPEED, the output has a colon separating the two rates, but when you issue SET TERMINAL/SPEED, you must separate the two rates with a comma. See the examples.

Rates are expressed in baud, or bits per second. A rate of 110 baud is the equivalent of 10 characters per second. For baud rates above 110, the rate divided by 10 is the equivalent in characters per second. Thus, 300 baud, the usual LA36 rate, is equivalent to 30 characters per second, and 9600 baud, the usual video terminal rate, is equivalent to 960 characters per second.

In any case, you are limited to the hardware speed settings for your terminal. See the documentation provided with your terminal, or your system manager, for further information.

/UPPERCASE

/NOLOWERCASE

Displays all characters typed on the terminal appearing in uppercase. The /UPPERCASE attribute is the default. /NOLOWERCASE is the same as /UPPERCASE.

/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

The following qualifiers set hardware characteristics of terminals. If you include SET TERMINAL/INQUIRE in your LOGIN.CMD file, you will probably never need to set any of these attributes because this command will have set them already. See the description of the /INQUIRE attribute in Group 1: Common Use section.

SET TERMINAL (Cont.)

For certain common models, you can set a number of characteristics by typing the terminal model as a qualifier for the SET TERMINAL command. For other terminals, however, you must either set characteristics explicitly or alter the terminal driver to recognize them. See the RSX-11M/M-PLUS I/O Drivers Reference Manual for complete information on the settings for each terminal model.

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 a terminal as to what it is.

Here are the models for which SET TERMINAL/model establishes a number of the terminal characteristics:

ASR33	ASR35	DTC01
KSR33	LA12	LA30P
LA30S	LA34	LA36
LA38	LA50	LA100
LA120	LA180S	LA210
LN03	LQP02	LQP03
VT05B	VT50	VT52
VT55	VT61	VT100
VT101	VT102	VT105
VT125	VT131	VT132
VT200_SERIES	PRO_SERIES	

Most of these are DIGITAL terminals. Those with "LA" designations are hardcopy terminals; those with "VT" designations are video models, as are the PRO-series terminals. Those with "SR" follow the Teletype¹ pattern.

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

Setting a terminal to a particular model designation automatically sets a number of attributes for the terminal. See the RSX-11M/M-PLUS I/O Drivers Reference Manual for complete information on the settings for each terminal model.

Note that you can also set these attributes individually. Nonprivileged users can set them for their own terminals; privileged users can set them for any terminal.

The defaults for these attributes depend on how your terminal is known to the system when you log in. You can find out how your terminal is known to the system by logging in and issuing a SHOW TERMINAL command.

1. Teletype is a trademark of the Teletype Corporation.

SET TERMINAL (Cont.)

/[NO]ADVANCED_VIDEO

States whether a VT100-series terminal includes the advanced video option, which gives it the capability to blink, bold, and flash parts of the screen, and to set the screen at 132 columns. See your system manager for more information. If the SET-UP blinks in setup mode, you have the advanced video option.

/[NO]ANSI_CRT

States whether the terminal output is a subset of the ANSI standard. Terminals with the /ANSI_CRT attribute do not output DIGITAL-private escape sequences.

/[NO]AUTOBAUD

The /AUTOBAUD qualifier enables autobaud detection on a remote dial-up line. This means that when a remote terminal dials up, the terminal driver samples the first character sent along the line to determine the incoming baud rate and sets the interface speed accordingly. The default is /NOAUTOBAUD.

/[NO]BLOCK_MODE

States whether the terminal is capable of local editing and block-mode transmission.

/CRFILL:n

Sets the attribute used to accommodate the carriage-return/line-feed process on hardcopy terminals. While the terminal is performing the line feed, a number of null characters may be needed to fill in for the time the terminal is not capable of accepting new characters. The value n can be any number from 0 through 7.

/[NO]DEC_CRT

States whether the terminal is upward-compatible with the VT100-series of terminals.

/[NO]EDIT_MODE

States whether the terminal can perform ANSI-defined advanced editing functions. The VT102 is such a terminal.

/[NO]FORMFEED

If the terminal hardware supports form feeds, the terminal should be set /FORMFEED. If form feeds are handled by the software providing line feeds, the terminal should be set /NOFORMFEED.

/HARDCOPY

/NOSCOPE

Sets terminal as a hardcopy terminal. /NOSCOPE is the equivalent of /HARDCOPY.

SET TERMINAL (Cont.)

/[NO]HOSTSYNC

Controls whether the terminal temporarily locks the keyboard when the typeahead buffer is full. /NOHOSTSYNC, the default, does not lock the keyboard.

The /HOSTSYNC qualifier temporarily locks the keyboard when the typeahead buffer is full. Locking the keyboard prevents you from entering information that the operating system is not ready to accept. Once the terminal transmits all of the information in the typeahead buffer to the operating system, it unlocks the keyboard to receive further information from you.

When you specify /HOSTSYNC, the operating system generates a CTRL/S to lock the keyboard and a CTRL/Q to unlock it.

The /HOSTSYNC qualifier is recommended for users with terminals that transmit information in blocks --- that is, DIGITAL terminals VT131 or VT132, or the terminals of any other vendor that emulate these two models. If you do not have such a terminal, it is unlikely that you will fill your terminal's typeahead buffer.

/LFFILL

Sets the attribute used to accommodate vertical spacing on terminals. While the terminal is performing a vertical tab, line feed, or form feed, four null characters are sent to fill in for the time the terminal is not capable of accepting new characters.

/MODEL:n

This parameter is used primarily to identify terminals that are not on the automatic setup list. Your system manager will tell you if you need to use this parameter.

SYSTEM MANAGERS

The argument n can be any decimal number from 0 through 255 (0-377(octal)). The first half of these numbers (0-127(decimal) or 0-177(octal)) is reserved for use by DIGITAL. Each of the terminals on the automatic setup list is assigned one of these numbers. The second half (128-255(decimal) or 200-377(octal)) is available for other terminals.

Setting a terminal with a /MODEL value has little effect by itself. If the /MODEL value is that of one of the terminals on the automatic setup list, the effect will be the same as setting the terminal to that model. If the /MODEL value is unused, then setting the terminal to that value affects nothing but the output from SHOW TERMINAL/MODEL.

If you wish automatic setup for those terminals not included in the automatic setup list, you can add /MODEL values to the table in SY0:[11,10]TTDAT.MAC and then reassemble and relink the terminal driver with the new table. Procedures for adding entries to the table are given in the comments in TTDAT.MAC.

SET TERMINAL (Cont.)

NOTE

If you do this, a SET TERMINAL/MODEL command with the proper value will set up such a terminal automatically. You cannot identify such terminals by a name; only a /MODEL value is permitted.

/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 hardcopy terminals. This attribute sets the length of the page. (Width is set through the SET DEVICE command, Section 5.11.2.)

/PRINTER_PORT

States whether the terminal has a port for attaching a line printer. For example, the VT200-series terminals have printer ports.

/[NO]REGIS

States whether the terminal supports the ReGIS graphics set. The VT125 and VT240 terminals support the ReGIS graphics set.

/SCOPE

/NOHARDCOPY

Sets terminal as a video terminal. /SCOPE is the equivalent of /NOHARDCOPY.

/[NO]SOFT_CHARACTERISTICS

States whether the terminal accepts software-defined character sets. The VT200-series terminals accept software-defined character sets.

/[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.

/[NO]TTSYNC

Controls whether the operating system responds to CTRL/S or CTRL/Q when you press these control characters on your terminal. The default, /TTSYNC, causes the operating system to respond to these characters.

CTRL/S delays output to your terminal; CTRL/Q resumes it. You usually use these two control characters alternately to control the flow of information on your terminal screen. Note that you do not lose information when you use CTRL/S and CTRL/Q.

Note also that if you set your terminal to /NOTTSYNC, the NO SCROLL key on the VT100-series terminals and the HOLD SCREEN key on the VT200-series terminals no longer work.

SET TERMINAL (Cont.)

Group 3: Task Setup

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. See the RSX-11M/M-PLUS I/O Drivers Reference Manual for information on how tasks can set terminal attributes.

/[NO]ECHO

Enables (or disables) echoing of characters typed on the terminal. The /ECHO qualifier is the default. Characters typed on the terminal are echoed on the terminal.

The /NOECHO qualifier suppresses the echo. (The system task LOGIN uses this feature to keep passwords secret.)

The /NOECHO qualifier changes nothing but echoing. Commands can still be passed to the system, but the system passes no echo back.

/[NO]EIGHT_BIT

Enables (or disables) a terminal's /EIGHT_BIT attribute.

The /NOEIGHT BIT qualifier is the default. DIGITAL software employs 7-bit ASCII (without parity checking).

The /EIGHT BIT qualifier 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.

/[NO]ESCAPE

Enables (or disables) recognition of escape sequences from a terminal. The /NOESCAPE qualifier is the default. When you press the ESC (SEL, ALTMODE) key, 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 (SEL, ALTMODE) key, it is interpreted as the beginning of an escape sequence. The ESC key 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]FULL_DUPLEX

Enables (or disables) full-duplex mode at a terminal. The /NOFULL DUPLEX qualifier is the default. The terminal cannot accept input and receive output at the same time. "Half-duplex" is the usual industry term for this attribute.

SET TERMINAL (Cont.)

The /FULL DUPLEX qualifier allows the terminal to accept input and receive output at the same time. This attribute is useful where echoing is not required, with split-screen applications, and when the terminal is communicating with some full-duplex device.

/INTERACTIVE
/NOPASSALL

Specifies that any characters typed on the terminal should be interpreted by the terminal driver before they are sent to a task. CTRL/A is accepted as a command to an interactive terminal. NOPASSALL is the same as INTERACTIVE. This is the default setting.

/LOCAL
/NOREMOTE

Specifies that the terminal is not attached to a dial-up line. /NOREMOTE is the same as LOCAL. This is the default setting.

/[NO]PARITY:arg
 ODD
 EVEN

Parity generation and checking is used to verify the transmission of data between the terminal and the computer. Verification is done by sending an extra bit with each character, which serves as a check that the character transmitted was received properly.

This qualifier enables, disables, and determines the type of parity checking for a given terminal line. When parity checking is enabled, an additional bit, called a parity bit, is generated and sent with each character output from the computer.

The state of the parity bit, either 0 or 1, is determined by the argument used with the command. If SET TERMINAL/PARITY:EVEN is specified, the state of the parity bit is selected to force the total number of bits set to "1" to be even. The analogous situation holds for odd parity selection. The terminal checks the state of the parity bit to see if it is correct. If not, a transmission error has occurred. The same parity bit generation is done by the terminal when a character is typed, and checked by the computer when it is received.

Note that the terminal hardware must be set to the same mode of parity generation and checking for correct data transmission to occur. On VT100-series and VT200-series terminals, this is done through the terminal's SET-UP mode.

SET TERM/PARITY attribute is not related to the /EIGHT_BIT attribute. If parity is selected, then an extra bit is physically transmitted with each character. Eight-bit only affects whether the eighth data bit should be cleared or left alone.

SET TERMINAL (Cont.)

`/PASSALL`
`/NOINTERACTIVE`

Specifies that characters sent by the terminal should not be interpreted by the terminal driver. Therefore, the task must interpret all characters. CTRL/A may be a command to a task, in this case, but will not affect the operating system. This parameter automatically sets the terminal to the `/EIGHT_BIT` attribute.

`/[NO]PASTHRU`

Controls whether the operating system executes special characters according to their original function, or ignores them and passes all data in their binary form to an application program.

The default is `/NOPASTHRU`, where the terminal keeps the original function of special characters.

The `/PASTHRU` qualifier allows you to suppress standard operating system responses to special characters, thereby controlling output to the terminal screen. Neither your terminal, nor any application programs running at your terminal, respond to special characters in the usual way. Your terminal does not recognize any control characters, except CTRL/S and CTRL/Q if your terminal is set to `/TTSYNC`. In addition, your terminal does not respond to carriage return and line feed characters.

If your terminal does not seem to work, it may have been inadvertently set to `/PASTHRU`.

The `/PASTHRU` qualifier is used primarily by applications programmers.

`/REMOTE`
`/NOLOCAL`

Identifies a terminal number with a dial-up line. A user can call in and attach a remote terminal. `/REMOTE` is the same as `/NOLOCAL`.

This keyword has nothing to do with the LOCAL/REMOTE switch found on some terminals, which sets the terminal as a typewriter (LOCAL) or terminal (REMOTE), nor does it have anything to do with DECnet host terminals.

`/[NO]SLAVE`

Enables (or disables) slave status at a terminal. The `/NOSLAVE` qualifier is the default. The terminal can both send unsolicited input and receive system output. The user has full access to all facilities of the terminal.

This is a rarely used parameter. Most tasks that need a slave terminal will attach the terminal without specific action by the user.

This parameter is not privileged, but if you set your terminal `/SLAVE`, you will need to go to a privileged terminal to set the terminal back to `/NOSLAVE`.

SET TERMINAL (Cont.)

/[NO]TYPEAHEAD

Enables or disables the terminal's typeahead attribute. The /TYPEAHEAD qualifier is the default. The terminal accepts input faster than it can send it. While waiting to be sent, this input is stored in the terminal's typeahead buffer, which holds 36 characters. This buffer size cannot be changed on RSX-11M; no numerical argument is accepted. If the buffer gets full, the terminal no longer accepts unsolicited input. When the buffer is full, the characters you type are no longer echoed, and the bell (buzzer, beeper) sounds each time you press a key. You can clear the typeahead buffer with a CTRL/X. /NOTYPEAHEAD disables this feature. The terminal driver cannot store characters and excess characters are lost.

/[NO]WRAP

Specifies that the terminal automatically wrap (or not wrap) lines longer than its line width. The /WRAP qualifier 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.

The /NOWRAP qualifier overrides this feature and permits unlimited line length.

Examples

```
$ SET (RET)
Function?  TERMINAL (RET)
Attribute? INQUIRE (RET)
```

The command in this example causes the system to test the terminal type and then set all appropriate attributes for that terminal type. In most cases, you will not need to change any attributes of your terminal after issuing this command.

```
$ SET TERMINAL/INQUIRE (RET)
```

This example is equivalent to the previous example. Include the command in this form in your LOGIN.CMD file and your terminal attributes are set each time you log in.

```
$ SET TERMINAL/VT100 (RET)
```

This example sets the issuing terminal as a VT100. The operating system "sees" the terminal as a VT100. If it is in fact a hardcopy terminal, output to the terminal may be garbled and the terminal may seem to jam. This command sets a number of defaults. These are shown in Table 3-2.

```
$ SET TERM/LA36 (RET)
```

This example sets the issuing terminal as an LA36. The operating system "sees" the terminal as an LA36. If it is in fact a video terminal, the only noticeable side effect will be the reprinting of characters deleted by the DELETE key. This command sets a number of defaults. These are shown in Table 3-2.

SET TERMINAL (Cont.)

```
$ SET TERM/SPEED:(110,300) (RET)
```

This example sets the issuing terminal to transmit at 110 baud and receive at 300 baud. The acceptable values for transmit and receive rates are inherent in the design of the terminal and other hardware. See the documentation provided with your terminal. After issuing this command, the user must reset the hardware to the same transmit and receive rates. You must issue the SET TERMINAL command first and then set the hardware so that the terminal driver can communicate with the terminal.

```
$ SET TERMINAL/LA36/SPEED:(300,300)/NOBROADCAST (RET)
```

This example sets the issuing terminal as an LA36, sets its speed to 300 baud for both transmit and receive, and inhibits broadcasting to the terminal.

Note

You can find how all attributes are set for your terminal with an unadorned SHOW TERMINAL command. See Section 3.8.2.

SHOW TERMINAL

3.8.2 SHOW TERMINAL

SHOW TERMINAL displays information about your terminal and the other terminals on your system.

Format

```
SHOW
Function?  TERMINAL[:Tnn:]

SHOW TERMINAL[:Tnn:] [/attribute] [/attribute[s]]
```

Attributes

/ADVANCED_VIDEO	/ANSI_CRT
/AUTOBAUD	/BLOCK_MODE
/BROADCAST	/CRFIL
/DCL	/DEC_CRT
/ECHO	/EDIT_MODE
/EIGHT_BIT	/ESCAPE
/FORM_FEED	/FULL_DUPLEX
/HARDCOPY	/HOLD_SCREEN
/HOSTSYNC	/INTERACTIVE
/LFFILL	/LOCAL
/LOGGED_ON	/LOWERCASE
/MCR	/MODEL
/PAGE_LENGTH	/PASSALL
/PASTHRU	/PRIVILEGED
/REGIS	/REMOTE
/SCOPE	/SLAVE
/SOFT_CHARACTERS	/SPEED
/TAB	/TTSYNC
/TYPEAHEAD	/UPPERCASE
/WRAP	

Terminal Types

/HT	/RT
/TI	/TT

Terminal Models

/ASR33	/ASR35
/DTC01	/KSR33
/LA12	/LA30P
/LA30S	/LA34
/LA36	/LA38
/LA50	/LA100
/LA120	/LA180S
/LA210	/LN03
/LQP02	/LQP03
/PRO SERIES	/VT05B
/VT50	/VT52
/VT55	/VT61
/VT100	/VT101
/VT102	/VT105
/VT125	/VT131
/VT132	/VT200_SERIES

SHOW TERMINAL (Cont.)

Command Option

TTnn:

Identifies the terminal about which you wish to display information. The default is the terminal from which the command is issued, which is TI:.

A SHOW TERMINAL command, such as the following:

```
$ SHOW TERMINAL (RET)
```

or

```
$ SHOW TERMINAL:TT56: (RET)
```

displays all the attributes set for your terminal or the terminal you name. See the examples.

SHOW TERMINAL/ALL displays information about all terminals on the system.

Attributes

Each SHOW TERMINAL attribute is directly related to a SET TERMINAL attribute. The meaning of each attribute is discussed in its place under SET TERMINAL in Section 3.8.1. In addition, each SHOW TERMINAL attribute can have a NO in front of it to allow you to determine which terminals do not have a particular attribute.

You can include more than one attribute in your SHOW TERMINAL command. This is the equivalent of entering a series of SHOW TERMINAL commands. That is, the following command:

```
$ SHOW TERMINAL/LA36/DCL (RET)
```

displays a list of all LA36s on your system and a list of all terminals set to DCL, but NOT a list of all LA36s set to DCL.

The displays returned by SHOW TERMINAL do not necessarily match the attribute names used in DCL. For instance, the following command:

```
$ SHOW TERMINAL/HARDCOPY (RET)
```

causes a display labeled NOCRT, meaning that the terminal is not a cathode-ray terminal.

The same display results from the following command:

```
$ SHOW TERMINAL/NOSCOPE (RET)
```

Table 3-2 shows the relationships between SHOW TERMINAL and its displays, and SET TERMINAL. For further information on the displays, see the examples.

SHOW TERMINAL (Cont.)

/HT
/RT
/TI:
/TT

These qualifiers display information about particular types of terminals on the system. /HT and /RT display a list of DECnet host terminals. /TI: is the same as SHOW TERMINAL without any qualifier. /TT displays a list of real terminals on the system.

If your SHOW TERMINAL command produces only a prompt and no output, there are no terminals on the system that match the attribute given in your command.

Table 3-2
Display of Terminal Attributes from SHOW TERMINAL

SHOW TERMINAL Attribute	SHOW TERMINAL Display	SET TERMINAL Attribute
ADVANCED_VIDEO	AVO	ADVANCED_VIDEO
ANSI_CRT	ANSI	ANSI_CRT
AUTOBAUD	ABAUD	AUTOBAUD
BLOCK_MODE	BLKMOD	BLOCK_MODE
BROADCAST	BRO	BROADCAST
CRFILL	HFILL=	CRFILL
DCL	CLI=DCL	DCL
DEC_CRT	DEC	DEC_CRT
ECHO	ECHO	ECHO
EDIT_MODE	EDIT	EDIT_MODE
EIGHT_BIT	EBC	EIGHT_BIT
ESCAPE	ESC	ESCAPE
FORM_FEED	FORM	FORM_FEED
FULL_DUPLEX	FDX	FULL_DUPLEX
HARDCOPY	NOCRT	HARDCOPY NOSCOPE
HOLD_SCREEN	HOLD	HOLD_SCREEN
HOSTSYNC	HSYNC	HOSTSYNC

(continued on next page)

TERMINAL OPERATIONS

SHOW TERMINAL (Cont.)

Table 3-2 (Cont.)
 Display of Terminal Attributes from SHOW TERMINAL

SHOW TERMINAL Attribute	SHOW TERMINAL Display	SET TERMINAL Attribute
INTERACTIVE	NORPA	INTERACTIVE NOPASSALL
LFFILL	VFILL	LFFILL
LOCAL	NOREMOTE	LOCAL NOREMOTE
LOWERCASE	LOWER	LOWERCASE
MCR	CLI=MCR	MCR
MODEL	TERM	MODEL
PAGE_LENGTH	LINES	PAGE_LENGTH
PASSALL	RPA	PASSALL INTERACTIVE
PASTHRU	PASTHRU	PASTHRU
PRINTER_PORT	PRINTER_PORT	PRINTER_PORT
PRIVILEGE	PRIV	PRIVILEGE
REGIS	REGIS	REGIS
REMOTE	REMOTE	REMOTE NOLOCAL
SCOPE	CRT	SCOPE NOHARDCOPY
SLAVE	SLAVE	SLAVE
SOFT_CHARACTERS	SOFT	SOFT_CHARACTERS
SPEED	SPEED	SPEED
TAB	HHT	TAB
TTSYNC	TTSYNC	TTSYNC
TYPEAHEAD	TYPEAHEAD	TYPEAHEAD
UPPERCASE	NOLOWER	UPPERCASE
WIDTH	BUF	WIDTH
WRAP	WRAP	WRAP

SHOW TERMINAL (Cont.)

Examples

```
$ SHOW (RET)
Function? TERMINAL (RET)
TT13:  [303,12]      [303,12]
        CLI  = DCL      BUF  = 132.      HFILL = 0      SPEED=(300,300)
        LINES = 66.     TERM = LA36      OWNER = SELF  BRO   NOABAUD
        LOWER NOPRIV  NOHOLD  NOSLAVE  NOESC  NOCRT  NOHFF  NOREMOTE
        ECHO  NOVFill NOHHT  NOFDX  WRAP  NORPA  NOEBC  TYPEAHEAD
        NOAVO NOANSI DEC   NOEDIT  NOREGIS NOSOFT NOBLKMOD
        NOHSYNC NOPASTHRU TTSYNC  NOPRINTERPORT
```

This example displays all the attributes for the user's own terminal. The command is the equivalent of SHOW TERMINAL:TI:. The attributes include the terminal number, the default DIRECTORY, the login UIC, plus all the attributes that can be set with SET TERMINAL. See Table 3-2 and the remaining examples.

```
$ SHOW TERMINAL:TT56: (RET)
TT56:  [7,100]      [303,5]
        CLI  = DCL      BUF  = 80.       HFILL = 0      SPEED=(9600,9600)
        LINES = 24.     TERM = VT100    OWNER = SELF  BRO   NOABAUD
        LOWER PRIV   NOHOLD  NOSLAVE  NOESC  CRT   NOHFF  NOREMOTE
        ECHO  NOVFill NOHHT  NOFDX  WRAP  NORPA  NOEBC  TYPEAHEAD
        AVO   ANSI  DEC   NOEDIT  NOREGIS NOSOFT NOBLKMOD
        NOHSYNC NOPASTHRU TTSYNC
```

This example displays all the attributes for another terminal.

```
$ SHO TERM/ESCAPE (RET)
$
```

This example shows that no terminal on the system is currently set to handle escape sequences.

```
$ SHOW TERMINAL/LA36/DCL (RET)
DCL=TT5:
LA36=CO0:
LA36=TT0:
LA36=TT34:
```

This example displays all the terminals on the system set as LA36s and all terminals on the system set to DCL. Note that the order of the display is not necessarily the same as the order the attributes were listed in. Note also that the effect is the same as a SHOW TERMINAL/LA36 followed by a SHOW TERMINAL/DCL. In other words, the display does not show all LA36s set to DCL.

SHOW TERMINAL (Cont.)

```
$ SHO TERM/LOWERCASE (RET)
LOWER=TT1:
LOWER=TT3:
```

```
.
.
.
```

```
$ SHO TERM/NOUPPERCASE (RET)
LOWER=TT1:
```

```
.
.
.
```

```
$ SHO TERM/NOLOWERCASE (RET)
NOLOWER=TT2:
NOLOWER=TT5:
```

```
.
.
.
```

```
$ SHO TERM/UPPERCASE (RET)
NOLOWER=TT2:
```

```
.
.
.
```

```
$ SHO TERM/HOLD_SCREEN (RET)
HOLD=TT21:
```

```
.
.
.
```

```
$ SHO TERM/NOHOLD_SCREEN (RET)
NOHOLD=TT1:
NOHOLD=TT2:
```

```
.
.
.
```

```
$ SHO TERM/MCR (RET)
MCR=TT14:
```

```
.
.
.
```

```
$ SHO TERM/DCL (RET)
DCL=TT20:
```

```
.
.
.
```

SHOW TERMINAL (Cont.)

```
$ SHO TERM/PRIVILEGED (RET)
PRIV=TT7:
```

```
.
.
.
```

```
$ SHO TERM/NOPRIVILEGED (RET)
NOPRIV=TT10:
```

```
.
.
.
```

```
$ SHO TERM/SPEED (RET)
SPEED=TT10:9600:9600
```

This example displays the speed settings for the terminal from which the command was issued.

Note that the transmission and reception rates are separated by a colon in this display. When you are setting these rates, the figures are separated by a comma and enclosed in parentheses.

```
$ SHO TERM/ASR33 (RET)
ASR33=TT22:
```

This example displays all the terminals on the system that are set to type ASR33. All the SHOW TERMINAL keywords naming specific models give the same kind of display.

If you use the NO option, the display lists NOASR33=TTn:.

```
$ SHO TERM/HARDCOPY (RET)
NOCRT=TT17:
```

```
.
.
.
```

```
$ SHO TERM/NOHARDCOPY (RET)
CRT=TT21:
```

```
.
.
.
```

```
$ SHO TERM/SCOPE (RET)
CRT=TT21:
```

```
.
.
.
```

```
$ SHO TERM/NOSCOPE (RET)
NOCRT=TT17:
```

CRT means cathode ray tube, another name for a video screen or scope.

SHOW TERMINAL (Cont.)

```
$ SHO TERM/CRFILL (RET)
HFILL=TT10:0
```

This example returns the /CRFILL value for the terminal from which it was issued.

```
$ SHO TERM/LFFILL (RET)
VFILL=TT42:
.
.
.
```

This example shows which terminal has its /LFFILL attribute enabled.

```
$ SHO TERM/FORMFEED (RET)
FORMFEED=TT13:
.
.
.
```

```
$ SHO TERM/NOFORMFEED (RET)
NOFORMFEED=TT15:
```

This example shows which terminals have their FORMFEED attribute enabled (or not).

```
$ SHO TERM/TAB (RET)
HHT=TT12:
.
.
.
```

```
$ SHO TERM/NOTAB (RET)
NOHHT=TT7:
```

This example shows which terminals have their TAB attribute enabled. HHT means Horizontal Hardware Tabs.

```
$ SHO TERM/PAGE_LENGTH (RET)
LINES=TT7:24
```

This example displays the page size (screen size) of the terminal from which the command was issued. A nonprivileged user cannot display the page size of another terminal with this command. Use the unadorned SHOW TERMINAL.

```
$ SHO TERM/ECHO (RET)
ECHO=TT1:
.
.
.
```

```
$ SHO TERM/NOECHO (RET)
NOECHO=TT15:
```

SHOW TERMINAL (Cont.)

```
$ SHO TERM/ESCAPE (RET)
ESCSEQ=TT1:
```

```
$ SHO TERM/NOESCAPE (RET)
NOESCSEQ=TT1:
```

.
.
.

ESCSEQ means ESCape SEquences.

```
$ SHO TERM/FULL_DUPLEX (RET)
FDX=TT2:
```

```
$ SHO TERM/NOFULL_DUPLEX (RET)
NOFDX=TT1:
```

.
.
.

FDX means full-duplex.

```
$ SHO TERM/PASSALL (RET)
RPA=TT1:
```

.
.
.

```
$ SHO TERM/INTERACTIVE (RET)
NORPA=TT3:
```

.
.
.

```
$ SHO TERM/NOPASSALL (RET)
NORPA=TT3:
```

.
.
.

```
$ SHO TERM/NOINTERACTIVE (RET)
RPA=TT1:
```

.
.
.

RPA means Read-Pass-All.

```
$ SHO TERM/REMOTE (RET)
REMOTE=TT1:
```


SHOW TERMINAL (Cont.)

```
$ SHO TERM/LOCAL (RET)
NOREMOTE=TT12:
```

```
.
.
.
```

```
$ SHO TERM/NOLOCAL (RET)
REMOTE=TT1:
```

```
$ SHO TERM/NOREMOTE (RET)
NOREMOTE=TT12:
```

```
.
.
.
```

This attribute has nothing to do with the LOCAL/REMOTE switch found on some terminals. Remote terminals access the system through dial-up lines while local terminals are hard-wired to the computer. This attribute has nothing to do with DECnet host terminals either.

The LOCAL/REMOTE switch on a terminal sets a terminal as a typewriter or terminal, respectively.

```
$ SHO TERM/SLAVE (RET)
SLAVE=TT56:
```

```
$ SHO TERM/NOSLAVE (RET)
NOSLAVE=TT1:
```

```
.
.
.
```

```
$ SHO TERM/TYPERHEAD (RET)
TYPERHEAD=TT1:
```

```
.
.
.
```

```
$ SHO TERM/NOTYPERHEAD (RET)
NOTYPERHEAD=TT27:
```

```
$ SHO TERM/WRAP (RET)
WRAP=TT1:
```

```
.
.
.
```

```
$ SHO TERM/NOWRAP (RET)
NOWRAP=TT45:
```

TERMINAL OPERATIONS

SHOW TERMINAL (Cont.)

```
$ SHO TERM/NOEIGHT_BIT (RET)
NOEBC=TT1:
```

```
·
·
·
```

```
$ SHO TERM/EIGHT_BIT (RET)
EBC=TT6:
```

EBC means eight-bit characters.

CHAPTER 4
HANDLING FILES

4.1 FILES ON RSX-11M SYSTEMS

The commands in this chapter are used to create files, list them in directories, remove them from directories, print copies of them on your terminal or the line printer, and alter their contents in various ways. Some error messages from commands described in this chapter are documented in Appendix A of this manual. Other error messages are documented in the RMS documentation supplied with your system.

All files are stored on volumes. These volumes are written on magnetic media, which can be placed on various physical devices. Once the volume is mounted, you can access the files on the volume. For more information about volumes and devices, see Chapter 5.

Within each volume, files are organized in directories. For the entire volume, there is a Master File Directory (MFD). The MFD is a file named [0,0]000000.DIR. Files in the MFD are all User File Directories (UFDs). The UFD files are named after the group and member numbers. Thus, UFD [200,1] is a file listed in the MFD and named [0,0]200001.DIR.

See Figure 4-1 for an illustration of the relationship between volume, MFD, and directory.

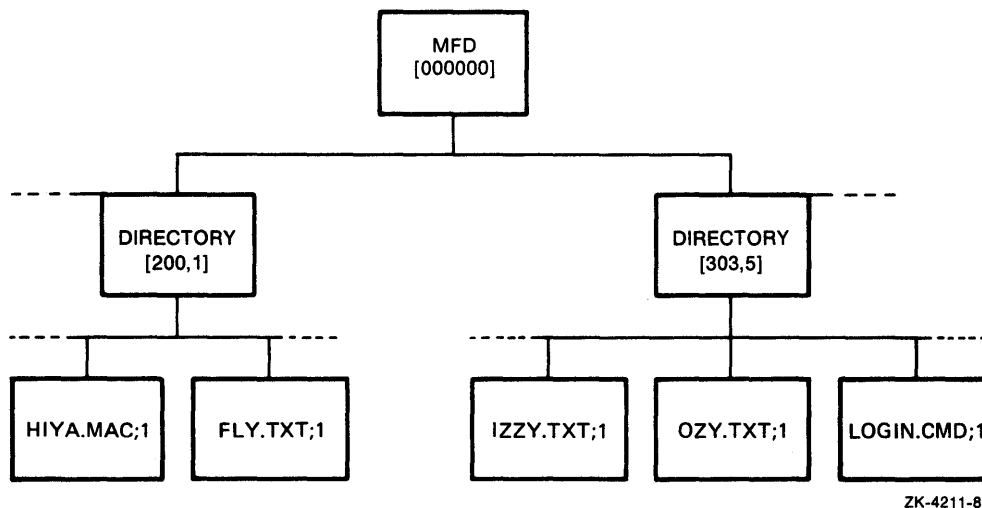


Figure 4-1 File Organization on a Volume

HANDLING FILES

4.1.1 File Ownership and Location

When you log in to the system, you identify yourself with a User Identification Code (UIC). You log in to your default device and directory. The default directory has the same group and member number as your UIC. Nonprivileged users can change their default directory, but not their UIC. Privileged users can change both. The UIC identifies the user; the directory identifies the location of the files.

If you issue the SHOW DEFAULT command, you can find out your default device (SY:), your default directory, and your UIC. This command is described in Section 4.5.2.

The default device is actually a volume mounted on a device. Most file-handling commands are directed to the volume and not the device, but both volume and device are identified in the same format, ddnn:, where dd is a 2-letter mnemonic for the device type and nn is the octal unit number of the device. The colon is a required terminator. Usually, the command description or function makes clear whether the command is directed to the physical device or to the volume.

Regardless of its physical designation, your default device is always called SY0:. This is a logical assignment made each time you log in. Information on your default device is kept by ACNT, the Account File Maintenance Program (Sections 3.1.1 and 3.1.3), and read from there each time you log in.

When you create a file, the system places the file name in a directory along with a User Identification Code (UIC), indicating the ownership of the file. Both the directory and the UIC have the same format, a 1- to 3-digit octal group number and a 1- to 3-digit octal member number separated by a comma and enclosed in brackets. These numbers range from 0 through 377 octal.

Directory entries consist of the names of files and pointers to file headers. The file header holds information about the file's owner and the location of the file on the mass-storage medium. Thus, a full file specification not only identifies the file, but locates it as well.

4.1.2 File Specification

A file specification uniquely identifies a file, indicating its location and contents. Many DCL commands require file specifications.

The format of a file specification is as follows:

```
ddnn:[g,m]name.typ;ver
```

ddnn:

Specifies the type of device and unit number on which the volume containing the file is mounted. It consists of a 2-letter device name followed by an optional 1- or 2-digit octal unit number terminated with a colon.

The 2-letter names for devices found on RSX-11M are listed in Section 5.1.

HANDLING FILES

[g,m]

Specifies the directory. g and m are octal numbers from 1 through 377. g is the number of your group and m is your member number. You usually have full access to the directories with the same group number as yours.

name

Specifies the name of the file, which can range in length from zero through nine of the following characters: the 26 letters A through Z and the numbers 0 through 9.

.typ

Specifies the file type. The type can be from zero through three of the following characters: the 26 letters A through Z and the numbers 0 through 9. In general, file type indicates the file contents. For example, .FTN is the file type for FORTRAN source programs.

Always separate the name from the file type with a period (.).

There are no restrictions on file types, but many system tasks use default file types for input and output files. These defaults and some system conventions on file types are summarized in Section 4.1.3. Null file types (0 length) are acceptable.

;ver

Specifies the version number, normally an octal number from 1 through 77777. The version number identifies different versions of the same file. When you create a file without specifying a version number, the system assigns the file version number 1. Each time you create a new version of the file, by editing it, for instance, the system adds one to the version number.

You must separate the file type and version number with a semicolon.

You can also create a file with an explicit version number.

If you create a file with version number 77777, you can be sure that file cannot be inadvertently superseded by a file with a higher number; 77777 is the highest possible version number. This is useful where you must have a particular version of a file for some operation.

In addition, you can name version 0 or version -1 in commands. Version 0 defaults to the most recent version of the named file; version -1 defaults to the oldest version of the named file.

In most cases, RSX-11M systems do not require you to enter the full file specification. Defaults are supplied for all fields of the file specification except the name (see Section 4.1.3). In addition, you can use wildcards to specify groups of related files (see Section 4.1.4).

HANDLING FILES

4.1.3 Defaults in File Specifications

Except for the file name, if you omit a field of the file specification, the system automatically supplies the field as specified here:

Field	Default
ddnn:	Your default device is established when you log in. SY0: is always your default device. Logging in assigns SY0: to some physical device. You can use SET DEFAULT to change the assignment of SY0:; you can display the assignment with SHOW DEFAULT. (See Section 4.5.) Remember that you do not have to use the unit number for devices numbered 0. DB: is the same as DB0:, for instance.
[g,m]	Your default directory is established when you log in. The default is the directory that has the same group and member numbers as your UIC. You can change the default directory with SET DEFAULT; you can display the default directory with SHOW DEFAULT. (See Section 4.5.)
.typ	Standard file types (see Table 4-1) are used as defaults. There is not a default file type in every situation.
;ver	For input files, the default is the highest numbered version; for output files, it is the next higher version number, or 1 if no previous version exists.

The defaults make it possible for you to specify files without having to type in the full file specification.

Suppose, for example, you log in with the UIC [200,20]. Your default device is DB0:, as recorded the System Account File, and assigned the name SY0:, and your default directory is [200,20]. Assume further that there are three sequentially numbered versions of a file called TEXT.TXT in the directory. If you wish to print the highest numbered version of this file (version 3) on your terminal, all of the following forms of the TYPE command produce the same desired result:

```
$ TYPE TEXT.TXT
$ TYPE TEXT.TXT;3
$ TYPE TEXT.TXT;0
$ TYPE [200,20]TEXT.TXT;3
$ TYPE DB0:TEXT.TXT
$ TYPE DB0:[200,20]TEXT.TXT
$ TYPE SY:TEXT.TXT
$ TYPE SY:[200,20]TEXT.TXT;3
$ TYPE SY:[200,20]TEXT.TXT;0
```

In most cases, you will use the first form, but if you wish to keep a permanent record of the terminal session, you may choose one of the more explicit forms. In any case, the system uses the final form given, as that form has all the defaults supplied. (Note that version 0 will not always be version 3, but is the highest numbered version, which is number 3 by definition in this example.)

HANDLING FILES

Table 4-1
File Types

File Type	Use
.BAS	BASIC-11 source program. System default.
.BLD	Indirect command file used by the SYSGEN procedure to create files needed to build system tasks. System default.
.B2S	BASIC-PLUS-2 source program. System default.
.CBL	COBOL-81 source program. System default.
.CMD	Indirect command file. System default.
.COR	SLP file used to correct a source file. System convention.
.DAT	File containing data, as opposed to code. System default.
.DIR	File is either Master File Directory or User File Directory. System default.
.DMP	Dump file created by File Dump Utility. System default.
.FTN	FORTTRAN-IV, FORTTRAN-IV-PLUS, or FORTTRAN-77 source program. System default.
.LST	Listing file. System default.
.MAC	PDP-11 MACRO-11 source program. System default.
.MAP	Task Builder map file. System default.
.MLB	Macro library. System default.
.OBJ	Object module output from assembler or compiler. System default.
.ODL	File containing Overlay Descriptor Language to be used by Task Builder. System default.
.OLB	Object module library. System default.
.PAT	Correction file used by assembler to create patched object module. Must be specified.
.PMD	File containing Postmortem Dump of interrupted task. System default. See the <u>RSX-11M/M-PLUS and Micro/RSX Task Builder Manual</u> for more information.
.POB	Patched object module input for PAT utility; must be specified. See Chapter 14 of the <u>RSX-11M/M-PLUS Utilities Manual</u> for more information on file types .PAT and .POB.

(continued on next page)

HANDLING FILES

Table 4-1 (Cont.)
File Types

File Type	Use
.SML	File containing system macro library. Other macro libraries have the type .MLB. System default, but must be specified on input file.
.STB	Symbol definition file created by Task Builder. System default. See Section 6.2 and the <u>RSX-11M/M-PLUS and Micro/RSX Task Builder Manual</u> for more information.
.SYS	Bootable system image. System default.
.TMP	Temporary file. System convention.
.TSK	Task image file. System default.
.TXT	Text file. System convention.
.ULB	Universal library. System default.

4.1.4 Wildcards in File Specifications

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

Simple wildcarding uses the asterisk (*) to replace all or any field in the file specification.

For instance, the following command:

```
$ DIRECTORY [303,*]
```

lists all files on the default volume with the group number 303, regardless of member number.

The following command:

```
$ DIRECTORY [*,5]
```

lists all files on the default volume with member number 5, regardless of group number.

The following command:

```
$ DIRECTORY [*,*]TEXT.TXT
```

lists the most recent versions of all files on the default volume named TEXT.TXT, regardless of the group or member number.

Likewise, the asterisk (*) in place of the version number means "all versions".

HANDLING FILES

The following command:

```
$ DIRECTORY WOM.DAT;*
```

lists all versions of the file WOM.DAT on the default volume and in the default directory.

The asterisk (*) can also be used to replace an entire file name or file type in much the same way.

The following command:

```
$ DIRECTORY *.DAT
```

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

The following command:

```
$ DIRECTORY COMMON.*
```

lists the most recent versions of all files with the name COMMON 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 asterisk (*) to replace an entire field in a file specification. Simple wildcarding works with all the commands in this chapter. The PRINT command accepts simple wildcarding. Many utilities also accept simple wildcarding. See the RSX-11M/M-PLUS Utilities Manual for information on how each utility handles wildcards.

For the following commands, DIRECTORY, DELETE, PURGE, COPY, RENAME, TYPE, APPEND, UNLOCK, and SET PROTECTION, a more elaborate form of wildcarding is available. In these commands, within file names and file types, the asterisk (*) can be used in a more complex manner. The asterisk actually means "match zero or all characters in this position."

Therefore, the following command:

```
$ DIRECTORY L*.TXT
```

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 UFD.

The following command:

```
$ DIRECTORY *L*.TXT
```

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 UFD.

The same substitutions can also be used in file types, so that the following command:

```
$ DIRECTORY SNOBLO.L*
```

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

You can use more than one wildcard in file names and file types.

HANDLING FILES

The following command:

```
$ DIRECTORY *F*D*.TXT
```

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, some commands permit the percent sign (%) to be used as a wildcard, but only within file names and file types. The percent sign means "match exactly one character in this position."

For instance, the following command:

```
$ DIRECTORY %.TXT
```

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

The following command:

```
$ DIRECTORY NOV%%85.TXT
```

lists all files with the type .TXT and a file name consisting of NOV and 85 separated by two characters on the default volume in the default directory.

The wildcards can be combined in a single file specification.

The following command:

```
$ DIRECTORY %L*T.TM%
```

lists all files whose names begin with a single character followed by an L and end with a T and with a file type consisting of .TM and another single character on the default volume in the default directory.

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.

4.1.5 Date-Related Qualifiers

The commands DIRECTORY, DELETE, PURGE, COPY, RENAME, TYPE, APPEND, UNLOCK, and SET PROTECTION accept several other qualifiers that add further flexibility to these commands.

First, there are the date-oriented qualifiers:

```
/DATE:dd-mmm-yy  
/SINCE:dd-mmm-yy  
/THROUGH:dd-mmm-yy  
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy  
/TODAY
```

These qualifiers all depend on the creation date of the file as shown in the DIRECTORY listing. The /DATE qualifier limits the operation of the command to files created on the specified date. The /SINCE qualifier limits the operation of the command to files created on or after the specified date. The /THROUGH qualifier limits the operation of the command to files created before or on the specified date. The /SINCE and the /THROUGH qualifiers can be combined to limit the

HANDLING FILES

operation of the command to files created within a given range of dates. The /TODAY qualifier limits the operation of the command to files created on the same day the command was issued.

You can enter the date in either of two forms:

dd-mmm-yy as in 25-MAY-85

or

mm/dd/yy as in 5/25/85

The system always displays dates in the first form.

These DCL commands also accept the following qualifier:

/EXCLUDE:filespec

The /EXCLUDE qualifier allows you to exclude a file or files from the operation of the command. The filespec argument to /EXCLUDE is a file specification, which must include a version number, but the version number can be an asterisk wildcard (*). Wildcards are accepted for all parts of the filespec argument to /EXCLUDE, but you cannot specify just wildcards in the filespec argument.

4.1.6 File Protection and Volume Protection

Data privacy and system security are provided by restrictions on access to volumes and the files on the volumes through volume protection and file protection. Volume protection refers to access to a mounted volume. You must have access rights to the volume before you can have access to any file on the volume. This access is determined through a protection code.

Once you have access to the volume, you must then have access to the file. File access is also determined through a protection code. The volume protection code and file protection code are slightly different because their functions are slightly different. The differences are noted in the following description.

Each file has a protection code, which specifies whether different kinds of system users who may access to the file, and what they may do to the file when they access it.

There are four kinds of users:

- SYSTEM - The operating system itself, and privileged users, those having group numbers of 10 or less.
- OWNER - The user having the same UIC as that the file was created under.
- GROUP - All users having the same group number as that the file was created under.
- WORLD - All other users.

There are also four kinds of access to files:

- READ - The user, or the user's tasks, may read, copy, print, or type the file, and if it is a task, run it.

HANDLING FILES

- WRITE - The user, or the user's tasks, may add new data to the file by writing to it.
- EXTEND - The user, or the user's tasks, may change the amount of disk space allocated to the file.
- DELETE - The user, or the user's tasks, may delete the file.

These forms of protection are expressed with a single character for each. The default protection applied to all files on the system that have not been otherwise protected is:

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

See also the discussion of directory protection in Section 4.2.2.

Volume protection codes differ from file protection codes in that EXTEND protection is called CREATE protection. The function of the code is the same. Either EXTEND protection or CREATE protection refers to the right to allocate space on a volume. CREATE under volume protection refers to your ability to allocate space for a new file, while EXTEND under file protection refers to your ability to allocate space within an existing file.

The default protection applied to all volumes on the system that have not been otherwise protected is:

```
SYSTEM:RWCD,OWNER:RWCD,GROUP:RWCD,WORLD:R
```

In fact, you can use a C in a file protection code or an E in a volume protection code and the command will be accepted, but it is good practice to use the correct protection category.

4.2 CREATING DIRECTORIES AND SEQUENTIAL FILES

The CREATE command without a qualifier enables you to create a sequential file directly at your terminal. The CREATE/DIRECTORY command permits a privileged user to create a User File Directory (UFD).

CREATE

4.2.1 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
File? filespec

CREATE filespec
```

Parameter

filespec

Specifies the name of the file to be created.

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

When you have finished entering text, type a CTRL/Z to close the file.

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

Examples

```
$ CREATE (RET)
File? COPY.CMD (RET)
; Command file to move files from Bob's directory to mine (RET)

        .ENABLE SUBSTITUTION (RET)
.LOOP: (RET)
        .ASKS FI What file (RET)
        COPY DB2:[303,26]'FI' 'FI' (RET)
        .GOTO LOOP (RET)
(CTRL/Z)
$
```

In this example, the user wanted to move a number of files from one directory to another. With CREATE, the user created an indirect command file at the terminal. After entering the CREATE command and file specification, the cursor (or print head) moves to the left margin. The terminal is attached for input to CREATE. Note that while CREATE is executing, the RETURN key acts only as a carriage return, and not as a means of entering commands to the system. No commands are accepted by DCL until after the CTRL/Z (echoed as ^Z).

```
$ CREATE JOE.TXT (RET)
Joe called at 4:30 on Monday. He'll be back from (RET)
Switzerland on Thursday, but he won't be in the office (RET)
until Friday afternoon. (CTRL/Z)
$
```

In this example, the user used CREATE to take notes on a phone call.

CREATE (Cont.)

```
$ CREATE CHURCHYLA.FEM (RET)
(CTRL/Z)
```

In this example, the user used CREATE to create an empty file for later use.

Notes

If you create a file without specifying a version number and no file of that name and type exists, the file is version 1. If a file of that name and type exists and you give no version number, the new file has a version number one higher than the highest numbered previous version.

If you specify a version number when creating the file, the system creates a file with the version number you give. If there is already a file with the same name, type, and version number, that file is deleted and replaced by the file you created.

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/U, CTRL/R, and the DELETE key are the only editing facilities available to you when creating sequential files at the terminal.

CREATE/DIRECTORY**4.2.2 CREATE/DIRECTORY**

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

Privileged users can create directories on volumes mounted on any device. Nonprivileged users can create directories only on volumes mounted on their own private (allocated) devices.

Format

```
CREATE/DIRECTORY[/qualifier[s]]
Device and UFD? [ddnn:][[g,m]]
```

```
CREATE/DIRECTORY[/qualifier[s]] [ddnn:][[g,m]]
```

Command Qualifiers

```
/ALLOCATION:n
/LABEL:volumelabel
/OWNER UIC:[uic]
/PROTECTION:(code)
/NOWARNINGS
```

Parameters

```
[ddnn:][[g,m]]
```

You must specify at least one of these parameters. If you specify one parameter, the default volume or directory is used for the other parameter. You cannot create a directory that matches both your default device and your default directory.

ddnn:

Specifies the device on which the volume you wish to create the directory is mounted. This parameter defaults to SY0:.

The volume must be a mounted, Files-11 volume.

```
[g,m]
```

Specifies the group and member numbers of the directory you wish to create. This parameter defaults to the current directory.

Command Qualifiers

```
/ALLOCATION:n
```

Specifies the number of directory entries (file names) for which space is to be allocated. n is rounded up to the next multiple of 32.

The default is /ALLOCATION:32.

```
/LABEL:volumelabel
```

Specifies that the volume label that you supply be compared with the label on the volume. If the names match, a UFD can be allocated. If they do not match, the command is rejected.

CREATE/DIRECTORY (Cont.)

If you do not specify this qualifier, the volume label is not checked.

/OWNER_UIC:[uic]

Specifies the User Identification Code (UIC) of the directory's owner. A UIC establishes the protection status of the directory. See Section 4.1.6 for more information.

The default is for the directory to be owned by you. 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 or her own.

/PROTECTION:(code)

Specifies the protection code for the directory file, not for files in the directory. Because the directory is a file, READ access is the right to list the directory, and WRITE access is the right to create files in the directory. If you do not have EXTEND access to a directory, you may be denied the right to create more than a certain number of files in the directory. DELETE access means the right to delete the directory file. If you delete a directory file, you must use the VFY utility to recover files from the directory. See Chapter 9 of the RSX-11M/M-PLUS Utilities Manual.

The default is the protection code the disk was mounted with. The default for mount protection is the protection code the volume was initialized with.

The default system protection code is as follows:

/PROTECTION:(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:R)

See also Section 4.1.6 for more information on file and volume protection.

/NOWARNINGS

Suppresses error messages resulting from the command.

Examples

```
$ CREATE/DIRECTORY (RET)
Device and UFD? DB2:[303,5] (RET)
```

This example creates directory [303,5] on the volume mounted on device DB2:. Nonprivileged users can create directories only on volumes mounted on a device they have allocated. Privileged users can create directories on volumes mounted on any device.

```
$ CREATE/DIRECTORY DB2:[303,5] (RET)
```

This example is identical to the previous one.

CREATE/DIRECTORY (Cont.)

```
$ CREATE/DIRECTORY (RET)
Device and UFD? [222,222] (RET)
```

This example creates UFD [222,222] on the default device.

```
$ CREATE/DIRECTORY (RET)
Device and UFD? DB6: (RET)
```

This example creates a directory on the volume mounted on DB6: with the same group and member number as the default directory.

```
$ CREATE/DIRECTORY/PROTECTION:(SYS:RWED,OWN:R,GRO:R,WORLD:R) (RET)
Device, UFD? DR2:[100,100] (RET)
```

This example creates directory [100,100] on the volume mounted on device DR2:. The protection code is the protection code applied to the actual file containing the directory, DR2:[0,0]100100.DIR. The code in this example permits privileged users to perform all operations on this file: READ (list the directory), WRITE (add files to the directory), EXTEND (alter the size of the directory file), and DELETE (delete the directory file). All other users can READ (list) the directory only. This protection mask has no relationship to the protection applied to files in the directory, or to volume protection, both of which may differ from the protection applied to the directory file. See Section 4.1.6.

4.2.3 EDIT

EDIT or EDIT[/EDT] invokes EDT, the DEC standard editor.

EDIT/SLP invokes the Source Language Input Program, a program maintenance utility. EDIT/EDI invokes EDI, the Line Text Editor. EDIT also invokes KED and K52, if present, and any unsupported editors you may have in your installation. The default is EDIT[/EDT].

EDIT/EDT

4.2.3.1 EDIT[/EDT]

EDIT invokes EDT, the DEC standard Editor. See the EDT Editor Manual for more information.

Format

```
EDIT[/EDT][/qualifier[s]] infile
  or
EDIT [/qualifier[s]] infile
```

Command Qualifiers

```
/[NO]COMMAND[:filespec]
/[NO]CREATE
/[NO]JOURNAL[:filespec]
/[NO]OUTPUT[:filespec]
/[NO]READ ONLY
/[NO]RECOVER
```

Parameter

infile

Specifies the file to be edited. If the file does not exist, EDT creates it.

You must supply a file name, but the type can be null.

Command Qualifiers

/[NO]COMMAND[:filespec]

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

The default is /COMMAND:EDTINI[.EDT].

If you use this qualifier and EDTINI.EDT or some other file you name does not exist, EDT issues no error message and continues with the editing session.

If you have a file EDTINI.EDT and do not wish to use it, use the /NOCOMMAND qualifier.

/[NO]CREATE

Controls whether EDT creates a new file when the specified input file is not found. Normally, EDT creates a new file to match the input file specification if it cannot find that file name in the specified directory. When you use the /NOCREATE qualifier in the EDT command line and accidentally type a file specification that does not exist, EDT prints an error message and returns you to the system command level.

EDIT/EDT (Cont.)**/[NO]JOURNAL[:filespec]**

Controls whether 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 file specification.

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 your inadvertently typing three CTRL/Zs in succession, the journal file is saved. In such a case, you invoke 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]

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, EDT creates version number 1. You can alter the name of the output file by including a file specification with the /OUTPUT qualifier. Otherwise, the qualifier need not be included.

If you specify /NOOUTPUT, you cannot exit EDT without including a file specification in your EDT EXIT command.

/[NO]READ_ONLY

Specifies whether you wish simply to read the file or edit it. If your command line includes /READ_ONLY, you can use the full facilities of EDT, but you cannot exit without including a file specification in your EDT EXIT command. Normally, you would use the 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 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 EDT to open the input file and then read 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 an EDT initialization file, you must specify the same file in the /RECOVER command line. In addition, if you specified a name for the journal file other than

EDIT/EDT (Cont.)

infile.JOU, you must include the /JOURNAL qualifier with the appropriate file specification. If journaling was not enabled on the original command line, you cannot recover the editing session.

Examples

```
$ EDIT (RET)
File? HORNBLOW.TXT (RET)
Input file does not exist
[EOB]
*
```

This example shows the EDIT command in its simplest form. The file HORNBLOW.TXT is created during the editing session.

```
$ EDIT HORNBLOW.TXT (RET)
```

This is the one-line form of the same command.

```
$ EDIT/OUTPUT:UMPIRE.MAC WEAVER.MAC (RET)
  1 .TITLE WEAVER
*
```

In this example, the user takes an existing file as input and edits it to create a new file.

```
$ EDIT/COMMAND:FORMAT RAMBLE.RNO (RET)
Input file does not exist
[EOB]
*
```

In this example, the user prepared an EDT initialization file called FORMAT.EDT. This file contains EDT command definitions designed to be used with a text formatting program. See next example.

```
$ EDIT/COMMAND:FORMAT/RECOVER RAMBLE.RNO (RET)
! This is FORMAT.EDT. Version 2.4   Ambrose Bierce, Maintainer
.
.
.
```

During the editing session started in the previous example, the system crashed. Once it was back running, the user duplicated his original command exactly, naming the same EDT initialization file and then added the /RECOVER qualifier. EDT begins reproducing the entire editing session, beginning with reading in the initialization file, the first line of which is seen in the example. All the editing commands and text entered during the session have been recorded in the file RAMBLE.JOU, which is also read back in by EDT. The result is that the entire editing session is repeated up to within a few keystrokes of the crash.

EDIT/EDT (Cont.)

If you wish to test the recovery procedure, start editing a file and then enter three CTRL/Zs as line-mode commands. This causes an unnatural exit from EDT, so the journal file is saved. You can then duplicate the original command line, add the /RECOVER qualifier, and watch the recovery of your previous editing session.

Notes

See the EDT Editor Manual for more information on journaling, initialization files, command and key definitions, and the other editing functions of EDT.

The qualifiers used with EDIT/EDT will not work with other forms of the EDIT command. If you use the /EDT qualifier, it must appear immediately after EDIT on the command line. Other qualifiers can float.

You can use the MCR command to invoke EDT in a different format that some users may find more convenient. See the EDT Editor Manual for the alternate command format.

EDIT/SLP

4.2.3.2 EDIT/SLP

EDIT/SLP invokes the Source Language Input Program (SLP), an editor designed for maintaining and updating source files. SLP enables you to delete, replace, or add lines to an existing file or create source files. SLP also has an optional audit trail which enables you to keep track of changes in the file. See the RSX-11M/M-PLUS Utilities Manual for more information on SLP.

Format

```
EDIT/SLP[/qualifier[s]] infile
```

Command Qualifiers

```
/[NO]AUDIT[:arg]
                POSITION:n
                SIZE:n
/[NO]CHECKSUM[:arg]
/[NO]LIST[:filespec]
/[NO]OUTPUT[:filespec]
/[NO]REPORT
/[NO]TAB
/[NO]TRUNCATE[:n]
```

Parameter

infile

The input file is the source program to be updated by SLP.

Qualifiers

```
/[NO]AUDIT[:(arg[s])]
                POSITION:n
                SIZE:n
```

Controls whether the output file includes an audit trail, and optionally allows you to specify the location and size of the audit trail. You can specify one or both of these values. If you specify only one, you can omit the parentheses, but the parentheses are required syntax if you specify both POSITION and SIZE. Separate the two arguments within the parentheses by a comma (,).

The POSITION: argument sets the starting position of the audit trail. The value of n can be from 0 through 132, representing the column at which the first character in the audit trail is to appear. This value is rounded up to the next highest tab stop. The default is to start the audit trail at column 80. Note that this default causes audit trails of more than a single character to wrap around when displayed on standard video terminals.

The SIZE: argument sets the length of the audit trail. The value of n can be from 0 through 14. The default is an audit trail of 8 characters.

The audit trail itself is defined from within SLP.

EDIT/SLP (Cont.)**/[NO]CHECKSUM[:arg]**

Controls whether a checksum is calculated for the SLP commands. If you specify checksum without an argument, SLP calculates the checksum value and prints it on your terminal. If you specify an argument, SLP calculates the checksum and compares it to what you have specified. If the numbers differ, a warning message is displayed, but the execution of SLP is not interrupted.

The default is /NOCHECKSUM.

/[NO]LIST[:filespec]

Creates a listing of a file with line numbers. If you do not give a file specification, the default file specification is filename.LST.

The default is /LIST. /NOLIST suppresses creation of the listing file.

/[NO]OUTPUT[:filespec]

Changes the name of the output file. The default output file specification is the same name and type as the input file and a version number one higher than the highest existing version of the file. If you do not wish to override this default, you do not need this qualifier.

The /NOOUTPUT qualifier suppresses the creation of an output file.

/[NO]REPORT

Controls whether line truncations that result from audit trails are reported. If you specify /REPORT, you receive warning messages on your terminal and the affected lines are marked with a question mark (?) in place of the period (.) in the line number in the listing file.

The default is /NOREPORT.

/[NO]TAB

Controls whether SLP replaces tabs or spaces at the end of each record containing an audit trail. If you specify /TAB, tabs are inserted. If you specify /NOTAB, spaces are inserted. The default is /NOTAB.

/[NO]TRUNCATE[:arg]

Requests SLP to truncate each record in the input file when it creates the output file. This qualifier allows you to delete an audit trail from a file previously updated with SLP. If you specify /TRUNCATE without a number, SLP truncates input records at the beginning position of the audit trail. If you specify a number, SLP truncates the records beginning at the column. The value of n can be from 0 through 132.

The default is /NOTRUNCATE.

EDIT/SLP (Cont.)

Notes

You must understand how to use SLP to use this command. See the RSX-11M/M-PLUS Utilities Manual for more information.

The qualifiers used with EDIT/SLP will not work with other forms of the EDIT command. The SLP qualifier must appear immediately after EDIT on the command line, but other qualifiers can float.

From MCR, you can invoke SLP in a different format that some may find more convenient. See the Utilities Manual for the alternate command format.

EDIT**4.2.3.3 EDIT/EDI and EDIT/editor**

EDIT/EDI invokes EDI, the Line Text Editor. The EDIT command can also be used to invoke KED and K52, which are supported but not bundled with RSX-11M systems as well as various unsupported editors.

Format

```
EDIT[/qualifier] [edit-input]
```

Command Qualifiers

```
/EDI
/USING:yyy
/SOS
/TECO
/MAKE
/MUNG
/KED
/K52
/OUTPUT:filespec
/CREATE
/[NO]READ_ONLY
```

Parameter**edit-input**

Any valid input to the chosen editor may be supplied here. Usually this will be a file specification, but anything, including a carriage return, that is valid for the chosen editor is a proper parameter for this command.

Command Qualifiers

Only one qualifier can be specified. This qualifier identifies the editor or function you wish to access.

/EDI

Specifies EDI, the Line Text Editor, an interactive, line-oriented editor used to create and maintain text and source files. See the RSX-11M/M-PLUS Utilities Manual for more information on EDI.

/USING:yyy

Specifies that an editor not supplied by DIGITAL is being used, where yyy is the installed name of the editor.

/SOS

Specifies the unsupported editor, Son of Stopgap (SOS).

```
/TECO
/MAKE
/MUNG
```

Specifies functions of the unsupported editor, Text Editor and Corrector (TECO).

EDIT (Cont.)

/KED
/K52

Invokes the Keypad Editor (KED), which is part of the Forms Management System (FMS-11) software. FMS-11 is not bundled with RSX-11M systems. The /K52 qualifier invokes a version of KED for VT52 terminals.

Other EDIT qualifiers for use with KED or K52 are /CREATE, /OUTPUT, and /[NO]READ_ONLY.

/CREATE

Use /CREATE with EDIT/KED or EDIT/K52 when you wish to edit a file that does not exist.

/OUTPUT:filespec

Use /OUTPUT with EDIT/KED or EDIT/K52. 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. You can alter the name of the output file by including a file specification with the /OUTPUT qualifier. Otherwise the qualifier is not needed.

If you specify /NOOUTPUT, no output file is created when you exit KED.

/[NO]READ_ONLY

Specifies whether you wish simply to read the file or to edit it. If your EDIT/KED or EDIT/K52 command includes /READ_ONLY, you can use the editor, but you cannot create a new version of the file. /NOREAD_ONLY is the default and need not be specified.

Examples

```
$ EDIT/EDI CHILI.MAC;2 (RET)
[00066 LINES READ IN]
[PAGE 1]
*
_
```

This example invokes EDI and names an existing input file.

```
$ EDIT/SOS HARRY.MAC;5 (RET)
Edit:DB0:[7,7]HARRY.MAC:5
*
```

This example invokes SOS for a file that already exists.

```
$ EDIT/SOS LONNNY.MAC (RET)
Input:DB0:[7,7]LONNNY.MAC;1
00100 _
```

This example invokes SOS for a file that is being created.

EDIT (Cont.)

```
$ EDIT/TECO PUNK.TST (RET)
```

*

This example invokes TECO for a file that already exists.

```
$ EDIT/MAKE JUNK.TST (RET)
```

*

This example invokes TECO for a file that does not exist.

```
$ EDIT/MUNG DOIT (RET)
```

Output depends on file.

This example invokes TECO where DOIT is a file of TECO commands.

Notes

There are no prompts from this form of the command.

No checking of the edit-input parameter is performed by DCL. The editor does all the checking.

The unsupported editors SOS and TECO are found on many RSX-11M systems. Commands to access these editors are included only for the convenience of users. No support is implied.

4.3 MAINTAINING A DIRECTORY

The following commands provide you with the means for keeping your directory up to date. Disk space on mass storage devices is an important resource. You should not keep files that you do not need.

DIRECTORY

4.3.1 DIRECTORY

DIRECTORY displays information on files in directories.

Format

```
DIRECTORY[/format-qual][/destination-qual] [filespec[s]]
```

Command Qualifiers

Format Qualifiers

```
/FULL
/BRIEF
/FREE[ddnn:]
/SUMMARY
/ATTRIBUTES
```

Destination Qualifier

```
/OUTPUT:filespec
```

Other Qualifiers

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

Parameter

filespec[s]

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

You can supply one or more file specifications, separated by commas, and directory information on the files you name 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 of another directory by supplying the directory name in this field. You can also specify device names in the form ddnn: in this field.

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

DIRECTORY (Cont.)**Command Qualifiers**

Command qualifiers are in three groups:

1. **Format qualifiers** control the appearance of the directory.
2. **Destination qualifiers** control where the command output is sent.
3. **Other qualifiers** modify the list of files by creation date or exclusion.

If you do not supply a format qualifier, the display is in standard format, giving the file name, type, and version number, the number of blocks the file occupies, and the date and time of creation.

If you do not include a destination qualifier, the display appears on your terminal.

If you do not include any of the other qualifiers, the display includes all files that otherwise qualify.

Format Qualifiers**/BRIEF**

Specifies that the display give file names, types, and version numbers only.

/FULL

Specifies that the complete directory entry be displayed, including File ID number, blocks used and allocated, the owning UIC, protection status of the file, in addition to all the information in the standard display.

/SUMMARY

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 [ddnn:]

Specifies that the display give the free space and number of free file headers on the default device or a specified device.

/ATTRIBUTES

Displays the Record Management Services (RMS-11) attributes of a file or files. This display includes the following:

- File name and type
- Creation date and time

DIRECTORY (Cont.)

- File organization
- Protection status
- Allocation information
- Last access date and time
- Record format
- Record size
- Primary and alternate key definitions for indexed files
- Bucket size for indexed and relative files

This qualifier invokes the RMS-11 DSP utility. Note that this display is produced for any file you specify in a DIRECTORY command, whether or not the file is an RMS-11 file. Some display information may not be meaningful for non-RMS files. See the RMS-11 documentation supplied with your system.

Destination Qualifier

/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

Specifies that you wish the DIRECTORY command to list only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish the DIRECTORY command to list only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you wish the DIRECTORY command to list only files created on or before the date given.

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

Specifies that you wish the DIRECTORY command to list only files created within that range.

/TODAY

Specifies that you wish the DIRECTORY command to list only files created on the same day the command is being issued.

DIRECTORY (Cont.)

/EXCLUDE:filespec

Specifies that you wish the DIRECTORY command not to list certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;* , TEST.*;* , or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

/REWIND

Causes the tape to rewind before starting the operation. This qualifier is for magnetic tapes only.

Examples

```
$ DIRECTORY (RET)
```

```
Directory DB1:[303,5]
1-FEB-85 14:16
```

A.A;1	1.		25-SEP-84 12:29
ANIML.BAS;1	6.		24-JAN-85 17:59
AZ.CMD;5	1.		02-APR-84 13:03
BYE.CMD;1	1.		25-SEP-84 12:29
CLEAN.CMD;1	1.		10-JAN-85 08:29
DUN.CMD;4	1.		23-JAN-85 08:05
EDT.CMD;22	1.		07-NOV-84 15:56
LOX.CMD;1	1.		27-DEC-84 10:21
LOGIN.CMD;6	1.		06-JAN-85 15:56
TI.CMD;4	1.		30-APR-84 08:36
.	.	.	.
F.TSK;1	4.	C	25-SEP-84 12:30
G.TSK;1	4.	C	25-SEP-84 12:30
PONG.TSK;2	12.	C	25-SEP-84 12:31
BUZZ.TXT;2	4.		12-JAN-85 09:13
IZZY.TXT;1	2.		06-MAR-84 14:33
JIVE.TXT;1	1.		16-JAN-85 13:23

```
Total of 2906./3043. Blocks in 160. Files
```

This example shows a directory listing in the standard format.

The device and directory are named at the head of the listing, along with the date and the time. All files in the directory are listed because the user did not supply any file specifications. The files shown with a C in column 3 are contiguous files, in this case, task images. You may see files in a directory with an L in this position, signifying that the files are locked. Files are locked when they are closed improperly. See Section 4.4.5 for information on the UNLOCK command.

DIRECTORY (Cont.)

\$ DIRECTORY/SUMMARY (RET)

Storage used/allocated for Directory DB1:[303,5]
1-FEB-85 14:15

Total of 2892./3033. blocks in 160. files

This example displays the output produced by the /SUMMARY qualifier. If the user had supplied one or more file specifications in the command, the summary would cover the blocks used and allocated for the specified files only.

\$ DIRECTORY/FREE (RET)

DB1: has 13742. blocks free, 3269280. blocks used out of 340670.
Largest contiguous space = 5173. blocks
124420 file headers are free, 131510. headers used out of 25593.

This example displays information about the default device.

\$ DIRECTORY/FREE LB0: (RET)

DB0: has 6074. blocks free, 334596. blocks used out of 340670.
Largest contiguous space = 3520. blocks
3297. file headers are free, 22296. headers used out of 25593.

This example displays information about the pseudo device LB0:. Notice that the display gives the physical device name for the pseudo device.

\$ DIR TI.CMD (RET)

Directory DB1:[303,5]
1-MAR-85 14:17

TI.CMD;4 1. 28-FEB-85 08:36

Total of 1./5. blocks in 1. file

This example displays information on the most recent version of the file named. See the next example.

\$ DIR TI.CMD;* (RET)

Directory DB1:[303,5]
1-MAY-85 14:17

TI.CMD;4 1. 30-APR-85 08:36
TI.CMD;3 1. 27-APR-85 16:32
TI.CMD;2 1. 27-APR-85 16:29
TI.CMD;1 1. 27-APR-85 16:28

Total of 4./20. blocks in 4. files

DIRECTORY (Cont.)

This example displays information on all versions of the file named because the user supplied a wildcard for the version number.

```
$ DIR/FULL TI.CMD (RET)
```

```
Directory DB1:[303,5]
1-MAY-85 14:17
```

```
TI.CMD;4 (300,56) 1./5. 30-APR-85 08:36 [303,5] [RWED,RWED,RWED,R]
```

```
Total of 1./5. blocks in 1. file
```

This example displays full information on the most recent version of the file named. The number in parentheses is the file identification number assigned by the system when the file is created. The first number is the file number, the second is the sequence number of the file. The UIC is the UIC of the owner, and the final column is the protection status of the file. See the notes.

Protection status is listed in the order system, owner, group, world. See Section 4.1.6 for an explanation of protection status and how it can be altered.

```
$ DIRECTORY/BRIEF *.CMD (RET)
```

```
Directory DB1:[303,5]
```

```
ADV.CMD;2
AZ.CMD;5
BYE.CMD;1
CLEAN.CMD;1
DAY.CMD;1
DTC.CMD;1
DUN.CMD;4
EDT.CMD;22
LOX.CMD;1
VTK.CMD;1
```

This example displays brief information on the most recent version of all files of the type .CMD in the default UFD.

```
$ DIRECTORY/TODAY (RET)
```

```
Directory DB2:[303,5]
5-MAY-85 13:44
Day of 5-MAY-85
```

```
05JAN85.LGI;1 0. 05-MAY-85 08:20
2051CH4.MEM;1 339. 05-MAY-85 10:11
NEWINSDC3.RNO;7 10. 05-MAY-85 10:05
2051CH4.MEM;2 320. 05-MAY-85 10:12
FUT.MAI;1 1. 05-MAY-85 13:05
NEWINSDC3.RNO;6 9. 05-MAY-85 09:55
NEWINSDC3.RNO;10 10. 05-MAY-85 10:39
NEWINSDC4.RNO;3 7. 05-MAY-85 11:46
FRK.CMD;6 1. 05-MAY-85 13:43
NEWINSDC4.RNO;4 11. 05-MAY-85 13:43
LOG.LOG;1 0. 05-MAY-85 13:43
```

```
Total of 708./736. blocks in 11. files
```

DIRECTORY (Cont.)

In this example, the user asked for a directory of all files created the same day the command was issued. Note that the directory listing includes the date. See next example.

```
$ DIRECTORY/TODAY/EXCLUDE:*.RNO;* (RET)
```

```
Directory DB2:[303,5]
```

```
5-MAY-85 13:45
```

```
*.RNO;* excluded
```

```
Day of 5-MAY-85
```

```
05JAN85.LGI;1      0.      05-MAY-85 08:20
2051CH4.MEM;1     339.    05-MAY-85 10:11
2051CH4.MEM;2     320.    05-MAY-85 10:12
FUT.MAI;1         1.      05-MAY-85 13:05
FRK.CMD;6         1.      05-MAY-85 13:43
LOG.LOG;1         2.      05-MAY-85 13:43
```

```
Total of 672./693. blocks in 6. files
```

In this example, the user listed all files created the day the command was issued but excluded all files with the .RNO file type. Note that the filespec argument to /EXCLUDE has a wildcard version number. Observe also that the exclusion is noted in the heading of the directory listing.

```
$ DIRECTORY/ATTRIBUTES YACHT.SEQ (RET)
```

```
SY:[303,5]YACHT.SEQ;1  FILE ORGANIZATION:      SEQUENTIAL
CREATED; 25-MAY-1985 12:32  REVISED: 15-JUN-1985 08:19 (5)
FILE PROTECTION:          [RWED,RWED,RWED,R]
RECORD FORMAT:            VARIABLE
RECORD ATTRIBUTES:       CARRIAGE RETURN
FILE ATTRIBUTES:
      ALLOCATION= 10  EXTEND QUANTITY=0
```

This example shows the display from the /ATTRIBUTES qualifier to DIRECTORY. Attributes can be displayed for any file, but the full display has meaning only for RMS-11 files.

```
$ DIR *.TSK, [301,370]*.TXT (RET)
```

```
Directory DB1:[303,5]
```

```
20-MAY-85 09:39
```

```
CLOCK.TSK;3      8.      C  15-APR-85 08:55
F.TSK;1          4.      C  25-APR-85 12:30
G.TSK;1          4.      C  25-APR-85 12:30
TICTAC.TSK;4     19.     C  01-APR-85 12:07
TREK.TSK;1       54.     C  25-MAR-85 12:31
```

```
Total of 89./89. blocks in 5. files
```

DIRECTORY (Cont.)

Directory DB1: [301,370]
20-MAY-85 09:39

FRONT.TXT;1	3.		16-MAY-85 11:23
EDITNEWS.TXT;12	25.		11-MAY-85 10:50
OZY.TXT;3	2.		11-MAY-85 10:50
TEXT.TXT;1	151.		11-MAY-85 10:50
NEW.TXT;1	0.	L	27-APR-85 14:36
AWARE.TXT;2	2.		27-APR-85 13:48
IZZY.TXT;10	2.		12-MAY-85 13:42

Total of 184./205. blocks in 7. files

Grand total of 273./294. blocks in 12. files in 2. directories

In this example, a user with the default directory of [303,5] requested directory information on files in two directories. The default directory was applied to the first files named, the *.TSK files; these were listed first in the display. The second set of files displayed, the *.TXT files, were from directory [301,370], which was explicitly specified in the command. Note the locked file. Compare this example with the next one.

\$ DIR [301,370]*.TXT, *.TSK (RET)

Directory DB1: [301,370]
20-MAY-85 09:40

FRONT.TXT;1	3.		16-MAY-85 11:23
EDITNEWS.TXT;12	25.		11-MAY-85 10:50
OZY.TXT;3	2.		11-MAY-85 10:50
TEXT.TXT;1	151.		11-MAY-85 10:50
NEW.TXT;1	0.	L	27-APR-85 14:36
AWARE.TXT;2	2.		27-APR-85 13:48
IZZY.TXT;10	2.		12-MAY-85 13:42
ADVENT.TSK;1	151.	C	13-MAY-85 14:12
DUNGEON.TSK;2	242.	C	21-APR-85 09:08
PONG.TSK;2	12.	C	23-MAR-85 14:13
YCLOCK.TSK;6	9.	C	08-APR-85 12:17
STAR.TSK;2	102.	C	13-JAN-85 09:07
TICTAC.TSK;5	19.	C	01-JAN-85 11:31
VTCHS.TSK;12	26.	C	06-FEB-85 15:39
XCLOCK.TSK;6	9.	C	03-MAY-85 16:48
LIFE.TSK;10	4.	C	03-APR-85 13:28

Total of 758./780. blocks in 16. files

The command in this example is quite similar to the command used in the previous one, but notice the difference in results. The same user, with default directory [303,5], issued the command but this time the directory [301,370] was specified first, for the *.TXT files. Even though no directory was specified for the *.TSK files, these files were also listed from directory [301,370] -- not from [303,5]. This demonstrates that specifying a directory in a DIRECTORY command resets the default directory for the rest of the command or until another directory is specified.

DIRECTORY (Cont.)

Notes

For more information on directories in general, see Chapter 3 of the RSX-11M/M-PLUS Utilities Manual for more information.

The full format for directory listings may include two dates. The first is the creation date of the file. The second date indicates the last time the file was opened by the system or a task, such as for backup. The decimal number in parentheses following the second date is the number of times the file has been opened in this fashion.

Your directory is a file that you own, but which is kept in directory [0,0] on the same volume as your directory. In directory [0,0] you will find the Master File Directory (MFD), which is a file named 000000.DIR. The MFD is the directory for directory [0,0]. All directories on the volume appear in this directory in the same form. The directory for [303,5] is called 303005.DIR; the directory for [7,40] is 007040.DIR, and so forth.

These directory files have file protection like all other files. The protection is for the directory file, not for files in the directory. Since the directory is a file, READ access is the right to list the directory, and WRITE access is the right to create files in the directory. If you do not have EXTEND access to a directory, you may be denied the right to create more than a certain number of files in the directory. DELETE access means the right to delete the directory file.

DELETE**4.3.2 DELETE**

DELETE deletes specified files and releases the storage space the files occupy.

Format

```
DELETE[/qualifier]
File(s)? filespec[s]

DELETE[/qualifier] filespec[s]
```

Command Qualifiers

```
/[NO]CONFIRM
/LOG
/[NO]QUERY
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yyy
/TODAY
/EXCLUDE:filespec
/NOWARNINGS
```

Parameter

filespec[s]

Specifies the file or files to be deleted.

You must supply the name, type, and version number fields of the file specifications of the files you wish to delete. Device and directory fields default to your current device and directory. You can use wildcards in any file specification field except the device field. You need not supply a file type to delete a file with a null file type.

You can only delete files to which you have DELETE access.

To name more than one file for deletion, separate their file specifications with commas.

Command Qualifiers

/LOG

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

```
/[NO]QUERY
/[NO]CONFIRM
```

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

```
Y - (Yes) Deletes file named and goes on to next file.
N - (No) Does not delete file named and goes on to next
file.
```

DELETE (Cont.)

- G - (Go) Deletes the file and goes on to delete all other files specified.
- Q - (Quit) Does not delete the file and exits the task. No additional files are deleted.

Remember that you can specify files by default or wildcard. See examples. If you do not specify a version number, /QUERY is the default. /[NO]CONFIRM is a synonym for /QUERY. See the examples.

/DATE:dd-mmm-yy

Specifies that you wish the DELETE command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish the DELETE command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you wish the DELETE command to affect only files created on or before the date given.

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

Specifies that you want the DELETE command to affect only files created within the specified range.

/TODAY

Specifies that you wish the DELETE command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the DELETE command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;* , TEST.*;* , or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

Examples

```
$ DELETE 
File(s)? *.DAT;* 
```

This example deletes all versions of all files of the type .DAT.

DELETE (Cont.)

```
$ DELETE FILE.FIL;1, ;3, .TXT;5 (RET)
```

This example deletes versions 1 and 3 of FILE.FIL, and version 5 of FILE.TXT. Note that no filename or type is specified for version 3, and no name for .TXT;5.

```
$ DELETE/QUERY (RET)
File(s)? *.TMP;* (RET)
Delete file DB1:[303,5]OGRE.TMP [Y/N/G/Q]? Y (RET)
Delete file DB1:[303,5]TROLL.TMP;1 [Y/N/G/Q]? Y (RET)
Delete file DB1:[303,5]ORC.TMP;1 [Y/N/G/Q]? Y (RET)
Delete file DB1:[303,5]ELF.TMP;1 [Y/N/G/Q]? N (RET)
Delete file DB1:[303,5]HOBBIT.TMP;1 [Y/N/G/Q]? N (RET)
Delete file DB1:[303,5]SNIPE.TMP;1 [Y/N/G/Q]? G (RET)
```

```
The following files have been deleted:
DB1:[303,5]SNIPE.TMP;1
DB1:[303,5]SNOPE.TMP;2
```

In this example, the user specified all files having the type .TMP in the DELETE command. Three files were deleted at the user's choice, and two were retained. The user then directed that all remaining files with the type .TMP be deleted. There were two more files, which were deleted and listed.

```
$ DELETE/CONFIRM (RET)
File(s)? *.TMP; (RET)
```

This example is equivalent to the previous one.

```
$ DELETE *.DOC (RET)
Delete file DB0:[303,5]WITCH.DOC;2 [Y/N/G/Q]? Y (RET)
Delete file DB0:[303,5]DRY.DOC;4 [Y/N/G/Q]? N (RET)
Delete file DB0:[303,5]PAYCHECK.DOC;1 [Y/N/G/Q]? Y (RET)
Delete file DB0:[303,5]LOADING.DOC;3 [Y/N/G/Q]? Y (RET)
Delete file DB0:[303,5]LOADING.DOC;4 [Y/N/G/Q]? Y (RET)
Delete file DB0:[303,5]LOADING.DOC;5 [Y/N/G/Q]? Y (RET)
Delete file DB0:[303,5]LOADING.DOC;6 [Y/N/G/Q]? N (RET)
```

In this example, the user specified the file type .DOC and a wildcard for the file name. Because no version number was given, the DELETE command defaulted to the /QUERY qualifier, enabling the user to choose which files to delete.

```
$ DELETE/LOG *.LST;* (RET)
```

```
The following files have been deleted:
DB1:[303,5]RANGER.LST;1
DB1:[303,5]TONTO.LST;1
DB1:[303,5]REID.LST;1
DB1:[303,5]SILVER.LST;1
DB1:[303,5]SCOUT.LST;1
DB1:[303,5]HORNET.LST;1
DB1:[303,5]KATO.LST;1
```


DELETE (Cont.)

In this example, the user specified all files with the type `.LST` in the `DELETE` command and asked that they be listed as they are deleted.

Notes

You must have `DELETE` access to delete a file.

If you wish to delete using wildcards, it is wise to get a directory listing using the same file specifications you plan to delete. In this way, you can be sure you are not deleting more files than you intend to delete.

Remember that under the default file protection on `RSX-11M` systems, you have `DELETE` access to your own files and all files in directories with the same group number.

See Chapter 9 for `DELETE` commands directed to the Queue Manager.

PURGE**4.3.3 PURGE**

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

Format

```
PURGE[/qualifier[s]]
File(s)? filespec[s]

PURGE[/qualifier[s]] filespec[s]
```

Command Qualifiers

```
/[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
```

Parameter

filespec[s]

Specifies the file group or file groups to be purged.

Because of the nature of the command, you cannot include a version number in the file specifications. Multiple file specifications must be separated by commas. Wildcards can be substituted for UFD, name, and type fields.

You can purge any files to which you have DELETE access.

Command Qualifiers

/[NO]LOG

Specifies that the files deleted by PURGE be listed on your terminal. The default is /NOLOG.

/KEEP:n

Specifies that the n latest versions of a file be retained.

If you do not use this qualifier, all versions but 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 version numbers are in numerical sequence and without missing numbers. See example.

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

PURGE (Cont.)**/DATE:dd-mmm-yy**

Specifies that you wish the PURGE command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish the PURGE command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you wish the PURGE command to affect only files created on or before the date given.

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

Specifies that you wish the PURGE command to affect only files created within that range.

/TODAY

Specifies that you wish the PURGE command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the PURGE command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

Examples

```
$ PURGE (RET)
File(s)? TEMPER.TSK (RET)
```

In this example, all versions of TEMPER.TSK but the latest are deleted.

```
$ DIRECTORY ASDIC.TM1;* , ELPASO.TEX;* (RET)
```

```
Directory DB1:[303,5]
20-MAY-85 13:44
```

```
ASDIC.TM1;1      1.      20-MAY-85 13:41
ASDIC.TM1;2      1.      20-MAY-85 13:41
ASDIC.TM1;3      1.      20-MAY-85 13:41
ASDIC.TM1;4      0.      20-MAY-85 13:41
ASDIC.TM1;5      1.      20-MAY-85 13:42
```

HANDLING FILES

PURGE (Cont.)

ELPASO.TEX;1	1.	20-MAY-85 13:42
ELPASO.TEX;2	1.	20-MAY-85 13:42
ELPASO.TEX;3	1.	20-MAY-85 13:42
ELPASO.TEX;5	1.	20-MAY-85 13:43
ELPASO.TEX;6	1.	20-MAY-85 13:43

Total of 9./45. blocks in 10. files

\$ PURGE/LOG/KEEP:3 ASDIC.TM1 (RET)

The following files have been deleted:

DB1:[303,5]ASDIC.TM1;1

DB1:[303,5]ASDIC.TM1;2

\$

\$

\$ PURGE/LOG/KEEP:3 ELPASO.TEX (RET)

The following files have been deleted:

DB1:[303,5]ELPASO.TEX;1

DB1:[303,5]ELPASO.TEX;2

DB1:[303,5]ELPASO.TEX;3

\$

\$

\$ DIRECTORY ASDIC.TM1;*, ELPASO.TEX;* (RET)

Directory DB1:[303,5]

20-MAY-85 13:46

ASDIC.TM1;3	1.	20-MAY-85 13:41
ASDIC.TM1;4	0.	20-MAY-85 13:41
ASDIC.TM1;5	1.	20-MAY-85 13:42
ELPASO.TEX;5	1.	20-MAY-85 13:43
ELPASO.TEX;6	1.	20-MAY-85 13:43

Total of 4./20. blocks in 5. files

In this example, the user started with two sets of five files. The five files named ASDIC.TM1 have version numbers in order. The five files named ELPASO.TEX are numbered 1,2,3,5,6. The user issued a PURGE command with the qualifier /KEEP:3. Versions 4, 5, and 6 of ASDIC.TM1 were kept, but only versions 5 and 6 of ELPASO.TEX. This is because the /KEEP:3 qualifier does not save the three highest numbered files, but rather the highest numbered file and the next two lower numbered files of the same name in numerical sequence. If there had been a file ELPASO.TEX;4, it would have been saved. Because there was none, the PURGE command task exited, its work done.

Notes

You can purge any file to which you have DELETE access.

COPY

4.3.4 COPY

COPY copies files.

COPY creates a sequential file copy of one or more sequential files, or of records with either indexed or relative file organization.

Format

```
COPY[/qualifier[s]]
From? infile[s][/qualifier]
To? outfile
```

```
COPY[/qualifier[s]] infile[s][/qualifier] outfile
```

Command Qualifiers

```
/ALLOCATION:n
/[NO]CONTIGUOUS
/NOWARNINGS
/OVERLAY
/PRESERVE DATE
/REWIND
/SHARED
/REPLACE
/OWN
/BLOCKSIZE:n
/DATE:dd-mmm-yy
/SINCE:dd-mmm-yy
/THROUGH:dd-mmm-yy
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
/TODAY
/EXCLUDE:filespec
```

Parameters

infile[s]

Specifies the input file or files to be copied.

You must have READ access to a file to copy it.

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 specify them.

outfile

Specifies a single output file to which the input file or files is copied.

You must have WRITE access to the directory to which you wish to send a copy. Under the default protection, you have WRITE access to your directory and to all directories with the same group number.

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 use a wildcard in either of these fields, you must use a wildcard in both.

COPY (Cont.)

The output file can be created by COPY. The output file need not exist when you issue the command. If a file of the same name and type already exists, then the file you create has a version number one higher than the highest existing version. 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 as well as to directories. See the examples.

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

Command Qualifiers**/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.

/[NO]CONTIGUOUS

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.

The default is /NOCONTIGUOUS.

/NOWARNINGS

Suppresses error messages resulting from the command.

/OVERLAY

Causes the contents of the input file or files to be copied into the output file. Whatever was in the output file 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 output file.

/REWIND

Causes the tape to rewind before starting the operation. This qualifier is for magnetic tapes only.

COPY (Cont.)**/SHARED**

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

/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.

/OWN

Changes the ownership of the file being copied from the source UIC to the destination UIC. After execution, both UICs own 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.

/BLOCKSIZE:n

Defines block size for magnetic tape. n can be octal or decimal; if decimal, you terminate n with a decimal point. For example:

/BLOCKSIZE:25.

/DATE:dd-mmm-yy

Specifies that you wish the COPY command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish the COPY command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you wish the COPY command to affect only files created on or before the date given.

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

Specifies that you wish the COPY command to affect only files created within that range.

/TODAY

Specifies that you wish the COPY command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the COPY command to affect certain files. The filespec argument to **/EXCLUDE** can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

COPY (Cont.)

Examples

```
$ COPY (RET)
From? [303,4]TSKBLD.CMD (RET)
To? TSKBLD.CMD (RET)
```

This example copies TSKBLD.CMD from [303,4] to the current directory and device. The file is still owned by the source UIC.

```
$ COPY *.BAS DK1: (RET)
```

This example copies all files with the file type .BAS from the current device and directory to the same directory on the volume mounted on device DK1:.

```
$ COPY/OWN TSKBLD.CMD [303,4]BLDFIL.CMD (RET)
```

This example copies TSKBLD.CMD from the current directory and device to [303,4], assigns ownership to the destination UIC, and also changes its name to BLDFIL.CMD.

```
$ COPY/CONTIGUOUS LITTER.DAT *.* (RET)
```

This example creates a new version of the file LITTER.DAT. The new version is contiguous.

```
$ COPY OLD1.FIL,OLD2.FIL (RET)
To? NEW.ONE (RET)
```

This example copies two previously existing files into one new file.

```
$ COPY CHARLA.DMP TT41: (RET)
```

This example prints a copy of CHARLA.DMP on TT41:.. This is a convenient means of sending messages longer than one line. If the terminal is busy at the time you send the copy, the copy is held until the terminal is clear and then sent.

```
$ COPY FLY.TXT SPIDER.TXT (RET)
```

This example creates the file SPIDER.TXT with the same contents as FLY.TXT, both on the current device and directory.

COPY (Cont.)

Notes

C is the short form of copy.

COPY does not affect file organization. If you wish to change file organization, use the CONVERT command, Section 4.3.6.

Usually, READ access is much broader than WRITE access. Under the default protection setup, your READ access covers the whole system, but your WRITE access is limited to other directories with the same group number. Thus, you can copy from many places that you cannot copy to.

Also, when you copy a file, you do not copy its protection code. Your copy has the default protection code. Use SET PROTECTION if you want to change the protection code of the copy.

Note that you can change the name, type, and version number of a single file as you copy it.

RENAME**4.3.5 RENAME**

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

Format

```
RENAME[/qualifier[s]]
Old file name? infile
New file name? outfile
RENAME[/qualifier[s]] infile outfile
```

Command Qualifiers

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

Parameters**infile**

Specifies the file to be renamed.

You may give a wildcard (*) for either the file name or the file type, or both. If you use a wildcard in these fields, you must supply an entry in the version number field. This may be a wildcard.

If you give a wildcard for version number, all versions retain their old version numbers. If no version number is supplied, only the highest version of the named file is renamed. It has the same version number as the old file. If other files having the new name exist, then you receive an error message.

outfile

Specifies the new name for the file.

Wildcards leave that portion of the file specification the same as before. No wildcard is needed for the version number.

Command Qualifiers

```
/DATE:dd-mmm-yy
```

Specifies that you wish the RENAME command to affect only files created on the date given.

```
/SINCE:dd-mmm-yy
```

Specifies that you wish the RENAME command to affect only files created on or since the date given.

RENAME (Cont.)**/THROUGH:dd-mmm-yy**

Specifies that you wish the RENAME command to affect only files created on or before the date given.

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

Specifies that you wish the RENAME command to affect only files created within that range.

/TODAY

Specifies that you wish the RENAME command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the RENAME command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;* , TEST.*;* , or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NEW_VERSION

Specifies that the renamed file be given a new version number.

/NOWARNINGS

Suppresses error messages resulting from the command.

Examples

```
$ RENAME (RET)
Old file name? INTRO.TXT (RET)
New file name? APPENDIX.TXT (RET)
```

In this example, the most recent version of INTRO.TXT becomes APPENDIX.TXT;1.

```
$ RENAME IZZY.TXT;4 FIZZY.* (RET)
```

In this example, IZZY.TXT;4 becomes FIZZY.TXT;1. Other versions of IZZY.TXT are not affected.

```
$ RENAME AMA.DOC;4 *.*;11 (RET)
```

In this example, AMA.DOC;4 becomes AMA.DOC;11.

```
$ RENAME MAIN.TSK;* EXTRA.TSK (RET)
```

In this example, all files named MAIN.TSK are renamed EXTRA.TSK. The version numbers remain the same, regardless of sequence and order of file creation.

RENAME (Cont.)

```
$ RENAME MAIN.TSK;* SUB.* (RET)
```

In this example, all versions of MAIN.TSK are renamed SUB.TSK. Versions are in the order of creation, with numbers unchanged.

```
$ REN EXHAUST.*;* REFRESH.* (RET)
```

In this example, all files named EXHAUST of whatever type are renamed REFRESH. Their file types remain the same. Note that a wildcard is given for the input file type. This makes an entry in the version number field mandatory.

Notes

You cannot rename files across devices.

Using wildcards is also tricky. Experiment before committing yourself.

CONVERT

4.3.6 CONVERT

The CONVERT command invokes the RMSCNV utility, which moves records from one file to another. RMSCNV reads records from an input file and writes them to an output file. The action of RMSCNV depends on the organization -- sequential, relative, or indexed -- of the two files, and on the qualifiers you include in the CONVERT command. See the RMS-11 documentation supplied with your system for more information.

Format

```

CONVERT [/qualifier[s]]
Input file? infilespec
Output file? outfilespec

CONVERT [/qualifier[s]] infilespec outfilespec

```

Command Qualifiers

```

/[NO]APPEND
/BLOCK_SIZE:n
/[NO]FIXED_CONTROL
/[NO]IDENTIFICATION
/INDEXED
/KEY[:n]
/[NO]LOG_FILE[:filespec]
/[NO]MASS_INSERT
/MERGE
/[NO]PAD[:[#]arg]
/RELATIVE
/[NO]REPLACE
/SEQUENTIAL
/[NO]TRUNCATE

```

Parameters

infile

Specifies the file that is the source of records to be written to the output file. CONVERT reads input records sequentially, regardless of the organization of the input file. No wildcards permitted.

outfile

Specifies the file that is to receive the records from the input file. The default file organization for the output file is sequential. If the output file is not sequential, use the /RELATIVE or /INDEXED qualifier to indicate the organization. If the output file is to be sequential, CONVERT can create the file; it need not exist prior to your entering the CONVERT command. CONVERT cannot create indexed or relative files; these must have been created prior to your entering the CONVERT command. See the RMS documentation supplied with your system. No wildcards permitted.

CONVERT (Cont.)**Command Qualifiers****/[NO]APPEND**

Specifies that you wish RMSCNV to append records to the end of an existing sequential file. If the output file is not sequential, RMSCNV ignores the qualifier. You cannot use both /APPEND and /REPLACE in the same command line.

The default is /NOAPPEND, but the action taken depends on the presence of the /REPLACE qualifier in the command line. If you specified /REPLACE, RMSCNV performs the replace operation. If you did not specify /REPLACE, RMSCNV creates the next higher version of the file.

This qualifier is the equivalent of the /AP switch in an RMSCNV command line.

/BLOCK_SIZE:n

Specifies the physical block size of an output file being created on magnetic tape. n can be from 18 through 8192 characters. RMSCNV rounds this value up to the next multiple of 4. If you do not specify this qualifier, the default tape block size is 512 characters.

This qualifier is the equivalent of the /BL switch in an RMSCNV command line.

/[NO]FIXED_CONTROL

The /FIXED_CONTROL qualifier directs RMSCNV to handle variable-with-fixed-control (VFC) format records in either the input file or the output file. If the fixed-control area of the input file and the output file are the same size, RMSCNV performs a straightforward copy. See the RMS-11 documentation supplied with your system for information on how RMSCNV handles other combinations. If you include this qualifier and neither file specifies VFC records, RMSCNV terminates.

This qualifier is the equivalent of the /WF switch in an RMSCNV command line.

The default is /NOFIXED_CONTROL. This means that if one of the named files contains VFC records, the fixed-control area of each record is ignored. That is, if the input file includes VFC records and the output file does not, only the variable portion of each record is written to the output file; if the output file includes VFC records and the input file does not, data is written only into the variable portion of each output record.

/[NO]IDENTIFICATION

Requests that RMSCNV print its current version number and patch level on your terminal. See the RMSCNV documentation for more information.

This qualifier is the equivalent of the /ID switch in an RMSCNV command line.

The default is /NOIDENTIFICATION.

CONVERT (Cont.)**/INDEXED**

The /INDEXED qualifier informs RMSCNV that the output is of indexed organization. Regardless of their organization, all input files are read sequentially.

RMSCNV reads each record from the input file, then applies the output file's record format, that is, key placement within the record, to the data. If you do not specify the /KEY qualifier with a value, the key of reference is the primary key; otherwise, it is the key you specify.

This qualifier is the equivalent of the /FO:IDX switch in an RMSCNV command line.

See the RMSCNV documentation for more information.

/KEY[:n]

Indicates the key that establishes the order in which records are read sequentially from an indexed input file and written to the output file. n can be from 0 through 9. The default is n=0 and indicates the primary key; n=1 is the first alternate key; n=9 is the ninth alternate key.

This qualifier is the equivalent of the /KR switch in an RMSCNV command line.

/[NO]LOG[:filespec]

Directs RMSCNV to summarize processing in a log. If you do not include a file specification, the log appears on your terminal. If you name a file, that file will be created and written to by RMSCNV. The log includes the following elements:

1. The command string in RMSCNV format. This will not be the same as DCL format.
2. Copies of all error messages produced during execution.
3. An indication of any duplicate-key problems. If the log is appearing on the terminal, the indication consists only of the following message:

SOME DUPLICATE RECORDS NOT WRITTEN

If the log is being written to a file, RMSCNV supplies the indicator DUP RCD= followed by the first 72 characters of the record that could not be written.

This qualifier is the equivalent of the /SL switch in an RMSCNV command line.

The default is /NOLOG meaning that only normal error messages appear on your terminal.

CONVERT (Cont.)**/[NO]MASS_INSERT**

Directs RMSCNV to activate the RMS-11 mass insert I/O technique and then use sequential put operations to insert records into the output file.

This is the equivalent of the /MA switch in an RMSCNV command line.

The default for nonsequential file organization is /NOMASS_INSERT. See the RMS-11 documentation supplied with your system for further information.

/MERGE

Directs RMSCNV to copy records from the input file into the output file. Both files must be of the same organization.

/[NO]PAD[:[#]arg]

Directs RMSCNV to pad records read from the input file to the output file's record length before writing them to the file. If you specify the qualifier without an argument, the pad character is null. If you do not include the number sign (#), the argument can be any printing ASCII character except: the number sign, question mark (?), or commercial-at sign (@). If you include the number sign (#), n can be an octal number from 0 through 377, representing the full ASCII character set. This enables you to use the three excluded characters as pad characters by specifying 43 for #, 77 for ?, and 100 for @.

Use this qualifier only when the output file specifies fixed-length records.

This qualifier is the equivalent of the /PD switch in an RMSCNV command line.

/RELATIVE

Informs RMSCNV that the output file is of relative organization. Regardless of their organization, all input files are read sequentially.

The /RELATIVE qualifier is the equivalent of the /FO:REL switch in an RMSCNV command line.

RMSCNV reads records from the input file and writes them into successive record cells of the output file, beginning with cell 1. If RMSCNV encounters a cell containing a record, it terminates with an error message. All records written to that point are in the output file. You should examine the two files to determine how far the processing went.

See the RMS-11 documentation supplied with your system for more information.

CONVERT (Cont.)**/[NO]REPLACE**

Directs RMSCNV to supersede an existing sequential file. RMSCNV replaces a file in the output account with the same file name, type, and version number. You cannot use /REPLACE and /APPEND in the same command line.

The /REPLACE qualifier is the equivalent of /SU in an RMSCNV command line.

The default is /NOREPLACE, but the default action depends on whether the command line includes an /APPEND or not. If you specify /APPEND, RMSCNV performs the append operation. If you do not specify /APPEND, RMSCNV creates the next higher version of the file.

/SEQUENTIAL

Informs RMSCNV that the output file is of sequential organization. Regardless of their organization, all input files are read sequentially. The default output file organization for RMSCNV is sequential.

This qualifier is the equivalent of the /FO:SEQ switch in an RMSCNV command line.

If you do not specify either the /APPEND or /REPLACE qualifier, and the output file is sequential, the output file need not exist before you issue the CONVERT command.

If the output file does not exist, RMSCNV creates an output file with the record attributes of the input file. RMSCNV then reads records from the input file and writes them sequentially into the new output file.

If the output file already exists, and the command does not include the /APPEND qualifier, RMSCNV creates the next higher version of the file. RMSCNV then reads records from the input file and writes them sequentially into the new version of the output file.

If the output file exists and you specify the /REPLACE qualifier, RMSCNV reads records from the input file and writes them sequentially into the output file, starting with the beginning of the file.

If the output file exists and you specify the /APPEND qualifier, RMSCNV reads records from the input file and writes them into the output file, starting with the record position following the last record already in the file.

/[NO]TRUNCATE

Directs RMSCNV to truncate records read from the input file to the output file's record length before writing them into the output file. The trailing bytes of the record are truncated.

CONVERT (Cont.)

The default is /NOTRUNCATE. If you do not specify /TRUNCATE and the input records are too long, RMSCNV terminates with the following message:

CON -- Input and output record sizes do not correspond

Examples

```
$ CONVERT (RET)
Input file? FOO.BAR (RET)
Output file? POOH.BAR (RET)
```

The output file does not exist. RMSCNV creates the sequential file POOH.BAR and copies the records from FOO.BAR into it.

```
$ CONVERT FOO.BAR POOH.BAR (RET)
```

This example is the equivalent of the previous one.

```
$ CONVERT/INDEXED GRIZZLY.BAR CROW.BAR (RET)
```

RMSCNV reads each record of the input file GRIZZLY.BAR sequentially and examines the contents of the primary key field, which is defined as an attribute of CROW.BAR. RMSCNV then inserts the record into CROW.BAR.

```
$ CONVERT/RELATIVE/KEY:1 SAND.BAR MAX.BAR (RET)
```

RMSCNV reads the indexed input file SAND.BAR, using the first alternate key of the file to establish the sequence of access. Then RMSCNV writes the records sequentially into an empty relative output file MAX.BAR, starting with record cell 1.

Note that the /RELATIVE qualifier refers to the organization of the output file and the /KEY qualifier refers to the organization of the input file.

```
$ CONVERT/APPEND/SEQUENTIAL/KEY:2 RUNNING.BAR CORNER.BAR (RET)
```

RMSCNV reads the indexed input file RUNNING.BAR, using the second alternate key of the file to establish the sequence of access. Then RMSCNV appends the records at the end of the existing sequential output file CORNER.BAR.

If the /APPEND qualifier had not been specified, RMSCNV would have created the next higher version of CORNER.BAR and written the records from RUNNING.BAR into it.

```
$ CONVERT/RELATIVE/FIXED_CONTROL MAMA.BAR PAPA.BAR (RET)
```

RMSCNV reads fixed-length records from the indexed input file MAMA.BAR, using (by default) the primary key of the file to establish the sequence of access. The RMSCNV writes the records sequentially in a VFC format into the existing relative file PAPA.BAR. Record format is an attribute of each file. As each record is written, the first bytes become the fixed control area.

CONVERT (Cont.)

```
$ CONVERT/TRUNCATE/BLOCK SIZE:1024/APPEND/KEY:9 BRER.BAR (RET)
Output file? MM3:BRER.BCK (RET)
```

RMSCNV reads the indexed input file BRER.BAR, using the file's ninth alternate key to establish the sequence of access. The RMSCNV writes the records to magnetic tape, truncating them to the output file's record length before adding them to the end of the existing file, and formatting the tape blocks to 1024 bytes each.

Notes

You must have a clear understanding of RMS-11 and the various forms of file organization it supports before using the CONVERT command. See the RMS-11 documentation supplied with your system for further information.

Sequential files used with the CONVERT command have been created by FCS.

4.4 OTHER FILE-RELATED COMMANDS

The commands in this section perform varied actions on your files. These commands are as follows:

TYPE	Displays files on your terminal.
PRINT	Prints files on the system line printer.
APPEND	Alter the contents of certain files.
DIFFERENCE	Displays the differences between two files.
UNLOCK	Unlocks locked files.

TYPE**4.4.1 TYPE**

TYPE prints selected files on your terminal.

Format

```
TYPE[/qualifier[s]]
File(s)? filespec[s]
```

```
TYPE[/qualifier[s]] filespec[s]
```

Command Qualifiers

```
/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
```

Parameter

filespec[s]

Specifies the file or files to be printed on your terminal.

You can specify any file to which you have READ access.

The file name and type must be specified explicitly or specified using wildcards (* or %). If no version is specified, the most recent version is printed. A wildcard in any field prints every file that matches otherwise. Both the * and the % wildcards are accepted.

You can also specify a device and a directory. If you do not, these fields default to the current device and directory.

Multiple file specifications must be separated by commas.

Command Qualifiers

```
/DATE:dd-mmm-yy
```

Specifies that you wish the TYPE command to affect only files created on the date given.

```
/SINCE:dd-mmm-yy
```

Specifies that you wish the TYPE command to affect only files created on or since the date given.

```
/THROUGH:dd-mmm-yy
```

Specifies that you wish the TYPE command to affect only files created on or before the date given.

TYPE (Cont.)

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

Specifies that you wish the TYPE command to affect only files created within that range.

/TODAY

Specifies that you wish the TYPE command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the TYPE command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;* , TEST.*;* , or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

/SHARED

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

Examples

```
$ TYPE (RET)
File(s)? LOGIN.CMD (RET)
.ENABLE QUIET
.IF <CLI> EQ "MCR" SET /DCL=TI:
SET TERMINAL LOWER
ASSIGN DB1: SY0:
@AZ
```

This example prints the file LOGIN.CMD from the current device and directory on your terminal.

```
$ TYP [303,12]LOGIN.CMD (RET)
.IF <CLI> EQ "MCR" ASN DB1:=SY:
.IF <CLI> EQ "DCL" ASSIGN DB1: SY0:
TYPE LOGIN.TXT
TYPE A.A
SET /DCL=TI:
```

This example prints the file LOGIN.CMD from the current device and directory [303,12] on your terminal.

PRINT

4.4.2 PRINT

PRINT queues files for printing on a line printer. It can also be used to queue jobs for other output devices. PRINT defines a QMG print job.

Format

```
PRINT[/qualifier[s]]
File(S) filespec[/qualifier[s]][, filespec[s]]

PRINT[/qualifier[s]] filespec[/qualifier[s]][, filespec[s]]
```

Command Qualifiers

```
/[NO]DELETE
/COPIES:n
/DEVICE:ddnn:
/[NO]HOLD
/JOB COUNT:n
/[NO]JOB PAGE
/[NO]LOWERCASE
/[NO]UPPERCASE
/QUEUE:queuename
/PAGE COUNT:n
/NAME:jobname
/PRIORITY:n
/FORMS:n
/LENGTH:n
/[NO]RESTART
/[NO]FLAG PAGE
/AFTER: (dd-mmm-yy hh:mm)
          (mn/dd/yy hh:mm)
/AFTER:TOMORROW
/NOWARNINGS
```

File Qualifiers

```
/[NO]DELETE
/COPIES:n
```

Parameter

filespec[s]

Specifies the file or files to be printed on the line printer.

Multiple file specifications must be separated by commas.

The asterisk wildcard (*) can be used in directory, name, type, and version number fields, but only to stand for the entire field. Wildcarding of the form L*.TXT is not accepted by PRINT, nor is the percent sign wildcard (%) accepted.

If your file specification includes no file type, the default file type is .LST.

Command Qualifiers

/[NO]DELETE

Specifies that files be deleted or not deleted from the directory after they are printed. The default is /NODELETE.

PRINT (Cont.)

The `/[NO]DELETE` qualifier to the file specifications can be used to override this qualifier for a given file in a list of file specifications.

`/COPIES:n`

Specifies the number of copies to be printed. The default is `/COPIES:1`. If your `PRINT` command includes this qualifier and specifies more than one file, the resulting QMG print job consists of the specified number of copies of the first file named, followed by that many copies of the second file, and so on. If you wish to print multiple copies of files by sets, use the `/JOB_COUNT` qualifier. See the examples.

The `/COPIES:n` qualifier to the file specification can be used to override this qualifier for a given file in a list of file specifications.

`/DEVICE:ddnn:`

Specifies that you want the file printed on a particular device.

`/[NO]HOLD`

Specifies that you wish the QMG print job to be held in its queue until you release it. See the `RELEASE/JOB` or `RELEASE/ENTRY` commands in Chapter 9, or the RSX-11M/M-PLUS Batch and Queue Operations Manual. The default is `/NOHOLD`.

`/JOB_COUNT:n`

Specifies the number of sets of files you wish printed. The default is `/JOB_COUNT:1`. If your `PRINT` command includes this qualifier and specifies more than one file, the resulting QMG print job consists of a set consisting of the files named in order, followed by another set, and so on. If you wish to print multiple copies of one or more files, but not in sets, use the `/COPIES` qualifier. See examples.

`/[NO]JOB_PAGE`

Specifies whether you wish a job flag page printed at the beginning of the QMG print job. The default is `/JOB_PAGE`. Use `/NOJOB_PAGE` when you are printing on special forms, or otherwise do not wish to include a job flag page. See the `/[NO]FLAG_PAGE` qualifier for controlling file flag pages. Figure 4-2 shows the difference between job flag pages and file flag pages.

`/QUEUE:queuname`

Specifies the name of the print queue in which the job is to be placed. The default is the `PRINT` queue.

`/[NO]UPPERCASE`

Specifies that the print job can be sent to a line printer without the lowercase character set. The default is `/UPPERCASE`.

01234567890123456789
01234567890123456789
01234567890123456789

** RSX-11M V4.1 **
** RSX-11M V4.1 **
** RSX-11M V4.1 **

DJ1:[7,67]TEST,CMD;1
COPY 1 OF 1
DELETION NOT SPECIFIED

24-MAY-85
24-MAY-85
24-MAY-85

09:39:45
09:39:45
09:39:45

01234567890123456789
01234567890123456789
01234567890123456789

```
TTTTTTTTT EEEEEEEEE SSSSSSS TTTTTTTTT  
TTTTTTTTT EEEEEEEEE SSSSSSS TTTTTTTTT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EEEEEEE SSSSSS  TT  
  TT      EEEEEEE SSSSSS  TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EE      SS      TT  
  TT      EEEEEEE SSSSSSS  TT  
  TT      EEEEEEE SSSSSSS  TT
```

....
....
....
....

```
CCCCCCCCC MM      MM DDDDDDD  ???  11  
CCCCCCCCC MM      MM DDDDDDD  ???  11  
CC          MMMM  MMMM DD      DD  ???  1111  
CC          MMMM  MMMM DD      DD  ???  1111  
CC          MM   M  MM DD      DD  ???  11  
CC          MM   M  MM DD      DD  ???  11  
CC          MM   MM DD      DD  ???  11  
CC          MM   MM DD      DD  ???  11  
CC          MM   MM DD      DD  ???  11  
CC          MM   MM DD      DD  ??  11  
CC          MM   MM DD      DD  ??  11  
CCCCCCCCC MM      MM DDDDDDD  ??  111111  
CCCCCCCCC MM      MM DDDDDDD  ??  111111
```

01234567890123456789
01234567890123456789
01234567890123456789

** RSX-11M V4.1 **
** RSX-11M V4.1 **
** RSX-11M V4.1 **

DJ1:[7,67]TEST,CMD;1
COPY 1 OF 1
DELETION NOT SPECIFIED

24-MAY-85
24-MAY-85
24-MAY-85

09:39:45
09:39:45
09:39:45

01234567890123456789
01234567890123456789
01234567890123456789

ZK-4210-85

PRINT (Cont.)

HANDLING FILES

Figure 4-3 File Flag Page

PRINT (Cont.)**/[NO]LOWERCASE**

Specifies that the print job must be sent to a line printer having the lowercase character set. The default is /NOLOWERCASE.

/PAGE_COUNT:n

Sets a limit on the number of pages the job may produce, ranging from 1 through 32767.

Without this qualifier, the number of pages is unlimited (n=0).

/NAME:jobname

Specifies a name for the print job. Your job name can be nine alphanumeric characters.

If you specify a job name, that name appears on the flag page at the beginning of the printed output. Otherwise, the file name of the first file is used as the job name.

The job name also appears in the SHOW QUEUE display. The job name includes the UIC under which it was entered.

There can be more than one job of the same name, but each job has a unique entry number which appears in the SHOW QUEUE display.

/PRIORITY:n

Sets the queue priority of the print job. For nonprivileged users, n can be from 0 through 150. Privileged users can set n up through 250. The default is 50.

The Queue Manager prints the highest priority jobs first. When two jobs have the same priority, the job that has been in the queue longest is printed first.

/FORMS:n

Specifies the type of form on which the job must be printed. n can have a value from 0 through 256. The default is 0. See your system manager for details on how to specify the particular forms used at your installation.

/LENGTH:n

Sets the length of a logical page; n can be any number from 0 through 255. The default is n=0. If you set a length, a form feed is automatically generated if one is not found within n lines.

When you set the length at 0, the default, the logical page length is unlimited.

This qualifier has no effect on how the printer handles form-feed characters. A form feed still moves the printer to the top of the next physical form, such as the perforation in a sheet of line-printer paper. This qualifier simply adds the requirement that the printer move to the top of the next physical page if no form-feed character is encountered within n lines. See the example using /LENGTH.

PRINT (Cont.)**/[NO]RESTART**

Controls requeuing of your job if it is stopped or aborted while being processed.

If /RESTART is specified, your job starts again from the beginning if it is aborted.

/NORESTART is the default. If the printer is stopped while your job is active, the job picks up where it left off when the printer is started.

In neither case are you missing any of your job, but with /RESTART your job is all in one piece.

If, however, the paper jams in the line printer, the operator must see that your job is restarted above where the jam began.

/[NO]FLAG_PAGE

Adds flag pages to each file in your print job. With /FLAG_PAGE, the number of flag pages is equal to the number of job flag pages that precede the job.

With /NOFLAG_PAGE (the default), your job is still preceded by a job flag page, but the files in the job are printed without any file flag page, separated only by form feeds. You can suppress the job flag page with the /NOJOB_PAGE qualifier. Figure 4-2 shows the difference between the job flag page and the file flag page.

**/AFTER: (dd-mm-yy hh:mm:[ss])
(mm/dd/yy hh:mm:[ss])**

Blocks the job until after the specified time. Depending on the print queues at that time, your job may be run immediately, or later, when its turn comes up in the queue.

The parentheses, hyphens (or slashes), colons, and the space between the calendar and clock fields are all necessary syntax. Note that a space is used only between the calendar and clock fields. The space is not necessary if either field is omitted.

You can specify the calendar field in either of two formats:

dd-mm-yy Uses a 1- or 2-digit number for the day, the first three letters for the month, and two numbers for the year.

mm/dd/yy Uses a 1- or 2-digit number for the month, a 1- or 2-digit number for the day, and two numbers for the year.

If you leave out the calendar field, the day defaults to the current date.

yy must be between 77 and 92. If you leave it out, the year defaults to the current year.

PRINT (Cont.)

hh:mm:ss is the hour, minute, and second.

If you leave out the clock field, the time defaults to 00:00:00 on the given date. The parameter ss need not be given. It defaults to 00. In fact, you cannot specify ss. If you do, it is checked for syntax only. The Queue Manager always takes a few seconds to start the job.

Specifying a time using /AFTER is equivalent to issuing a Print command at that time. The job may still have to wait in its queue to be printed.

/AFTER:TOMORROW

Specifies that you want the command to execute after midnight. Use this qualifier to queue print jobs for the early morning hours. If you have a command file that you run every day, you can include PRINT/AFTER:TOMORROW in the command file. This qualifier circumvents the problem that it is always after midnight.

/NOWARNINGS

Suppresses error messages resulting from the command..

File Qualifiers

These qualifiers are used to override the corresponding command qualifiers when you have given multiple file specifications as parameters to the PRINT command.

/COPIES:n

Overrides the /COPIES command qualifier when attached to a particular file specification.

/[NO]DELETE

Overrides the /[NO]DELETE command qualifier when attached to a particular file specification.

Examples

```
$ PRINT (RET)
File(s)? PASHA.CMD (RET)
PRI -- Job 305, name "PASHA", submitted to queue "PRINT "
```

This example prints the most recent version of PASHA.CMD from the default device and directory on the line printer. The Queue Manager returns a message confirming that the job has been submitted and given an entry number. You can use this entry number in QMG commands. See Chapter 9, and also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

```
$ PRINT/COPIES:20 (RET)
File(s)? PASHA.CMD (RET)
PRI -- Job 706, name "PASHA", submitted to queue "PRINT "
```

PRINT (Cont.)

This example prints 20 copies of PASHA.CMD. They are printed end-to-end with no flag page except at the beginning.

```
$ PRI/COP:20/FLAG_PAGE PASHA.CMD (RET)
PRI -- Job 321, name "PASHA", submitted to queue "PRINT "
```

This example prints 20 copies of PASHA.CMD, each with a flag page.

```
$ PRIN/COPIES:2 DAVID.TXT, DAN.TXT (RET)
PRI -- Job 18, name "DAVID", submitted to queue "PRINT "
```

This example prints two copies of DAVID.TXT followed by two copies of DAN.TXT. It does not print two sets of one copy of each file. If you wish to print sets of files, see the next example.

```
$ PRINT/JOB COUNT:2 DAVID.TXT, DAN.TXT (RET)
PRI -- JOB 311, NAME "DAVID", submitted to queue "PRINT "
```

This example prints a set consisting of a copy of DAVID.TXT followed by a copy of DAN.TXT, and then prints another set of DAVID.TXT followed by DAN.TXT.

```
$ PRI/FLAG PASHA.CMD, RAJAH.CMD, SHEIK.CMD (RET)
PRI -- Job 999, name "PASHA", submitted to queue "PRINT "
```

This example prints one copy of each file, each with its own flag page.

```
$ PRI/NAME:ARABY/FLAG PASHA.CMD, RAJAH.CMD, SHEIK.CMD (RET)
PRI -- JOB 805, NAME "ARABY", submitted to queue "PRINT "
```

This example prints one copy of each file, each with its own flag page, and gives the name ARABY to the job as a whole. The name ARABY appears in the print queue and at the head of the entire printed output.

```
$ PRINT/LENGTH:60 GLADRAG.MAC (RET)
PRI -- Job 22, name "GLADRAG", submitted to queue "PRINT "
```

This example prints a single copy of the file GLADRAG.MAC. This is a source program file without form-feed characters in it. The user wants to print it on line-printer paper, a physical form 66 lines long. To avoid printing over the perforations, the user has specified a logical length of 60 lines. Each time 60 lines have been printed, the logical page is complete and the printer moves to the top of the next physical page.

If the file had form feeds in it, the printer would move to the top of a physical page each time it encountered a form feed and each time it printed 60 lines without encountering a form feed.

PRINT (Cont.)

```
$ PRI/COPIES:20 MOB.COR, RIFRAF.COR, RABBLE.COR/COP:19, PROLE.COR (RET)
PRI -- Job 77, name "MOB      ", submitted to queue "PRINT "
```

This example prints 20 copies each of MOB.COR, and RIFRAF.COR, 19 copies of RABBLE.COR, and then 20 copies of PROLE.COR.

```
$ PRI/DELETE LASER.TST, PHASER.TST, RAZOR.TST/NODELE (RET)
PRI -- Job 406, name "LASER   ", submitted to queue "PRINT "
```

This example prints the three files and deletes the first two from the current UFD.

Only the versions specified in the PRINT command are deleted, in this case, the most recent versions of LASER.TST and PHASER.TST.

```
$ PRI/AFTER:(4-JUL-85 13:13) CONST.CON (RET)
PRI -- Job 809, name "CONST   ", submitted to queue "PRINT "
```

This example blocks the job in its queue until the specified date and time. The file itself remains in its directory. Deleting the file from its directory does not take it out of the queue, but does prevent it from being printed.

```
$ PRI/AFTER:(17:00) GUNGA.DIN (RET)
PRI -- Job 765, name "GUNGA   ", submitted to queue "PRINT "
```

This example prints the file after 5 P.M. on the day the command is entered. If the command is entered after the specified time, the file is printed immediately.

Notes

P is the short form of PRINT.

The PRINT command does not accept the percent sign wildcard (%). The asterisk wildcard (*) can only be used to substitute for a complete part of the file specification and NOT for part of a file name or file type as some commands do.

For more information on the PRINT command, see the RSX-11M/M-PLUS Batch and Queue Operations Manual. This manual also includes a description of the SHOW QUEUE command, which displays information about print jobs in queues.

Once your job is in a queue, you can modify some job attributes with SET QUEUE. See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

The Queue Manager automatically deletes .PMT and .DMP files after they are printed.

The /LOWERCASE and /UPPERCASE qualifiers have no effect on the nature of your output. They do not cause your job to be printed all uppercase or in lowercase. Rather, these qualifiers must match the attributes given to the printer when it was initialized. A printer without the lowercase character set can be initialized /LOWERCASE and accept print jobs with the /LOWERCASE qualifier on the PRINT command. This might be done if

PRINT (Cont.)

a printer with the lowercase character set was temporarily unavailable. In most cases, only printers with the lowercase character set are initialized /LOWERCASE, however.

Despite your setting of the /[NO]RESTART qualifier, your job may be restarted by commands to the Queue Manager from your system manager. See your system manager if you have jobs where restarting or not restarting may be a problem.

The flag pages and the job pages have different formats for the banner text. The job name page shows the UIC the PRINT command was issued under on the first line and a 9-character job name (derived from an explicit job name or the file name of the first file in the job) on the second line. The flag page shows the full 9-character file name on the first line and the file type and version number on the second line. See Figures 4-2 and 4-3.

Entry numbers run from 1 through 999 and are unique. Various QMG commands permit you to identify your job either by this number or by job name. The job name includes your UIC. See Chapter 9 and also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

APPEND

4.4.3 APPEND

APPEND appends to an existing sequential file records from one or more sequential files.

Format

```
APPEND[/qualifier[s]]
File(s)? infile[s]
To? outfile
```

```
APPEND[/qualifier[s]] infile[s] outfile
```

Command Qualifiers

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

Parameters

infile[s]

Specifies the file or files to be appended to the output file. You can use APPEND only on sequential files.

outfile

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 output file specifications.

There are no qualifiers for the output file specification.

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

Command Qualifiers

/DATE:dd-mmm-yy

Specifies that you wish the APPEND command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish the APPEND command to affect only files created on or since the date given.

APPEND (Cont.)

/THROUGH:dd-mmm-yy

Specifies that you wish the APPEND command to affect only files created on or before the date given.

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

Specifies that you wish the APPEND command to affect only files created within that range.

/TODAY

Specifies that you wish the APPEND command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the APPEND command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;* , TEST.*;* , or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppress error messages resulting from the command.

/REWIND

Causes the tape to rewind before starting the operation. This qualifier is for magnetic tapes only.

/SHARED

Specifies that you want others to be able to access the file while your append operation is under way.

Examples

```
$ APPEND (RET)
File(s)? BIRD.DOG;* (RET)
To? JOKER.TXT (RET)
```

This example appends copies of all versions of BIRD.DOG in order to the end of the most recent version of JOKER.TXT.

```
$ APPEND *.LST LIST.FIL (RET)
```

Appends copies of most recent versions of all files of the type .LST to LIST.FIL. Appended files appear in the order they are listed in the directory.

Notes

APPEND does not delete the input files. They remain intact. Copies of the input files are placed at the end of the output file.

DIFFERENCES

4.4.4 DIFFERENCES

DIFFERENCES compares two ASCII (text) files line by line to determine if parallel records (lines) are identical, and produces a listing of the differences between the files, if any.

Format

```
DIFFERENCES[/qualifier[s]]
Filespec1? infile1
Filespec2? infile2
```

```
DIFFERENCES[/qualifier[s]] infile1 infile2
```

Command Qualifiers

```
/CHANGE_BAR[:n]
/IGNORE:(arg[,s])
    BLANK_LINES
    COMMENTS
    FORM FEEDS
    SPACING
    TRAILING_BLANKS
/LINES:n
/NONUMBERS
/OUTPUT:filespec
/SLP[:au]
```

Parameters

infile1

Specifies the first of two files to be compared. The file name must be included. The default file type is .MAC.

infile2

Specifies the second of two files to be compared. The file name must be included. The default file type is .MAC

Command Qualifiers

/CHANGE_BAR[:nnn]

Specifies that the output consists of a listing of infile2 with change bars applied to each line in infile2 that does not have a corresponding line in infile1. The nnn argument is the number of the ASCII character to be used for the change bar. The default is 041, the exclamation point (!). Here are the codes for some commonly used ASCII characters:

!	041
#	043
\$	044
*	052
+	053
<	074
>	076
?	077
	074

DIFFERENCES (Cont.)

```

/IGNORE:(arg[,s])
    BLANK_LINES
    COMMENTS
    FORM_FEEDS
    SPACING
    TRAILING_BLANKS

```

Specifies that you wish certain defaults to be ignored in making the comparison. Without the /IGNORE qualifier, the comparison is strictly line-by-line. If you specify only one argument, you do not need the parentheses. If you specify more than one argument, enclose them in parentheses and separate them with commas (,).

The default is to include all these factors in the comparison.

The BLANK_LINES argument specifies that blank lines are to be excluded from comparison. The default is to include blank lines in the comparison.

The COMMENTS argument specifies that comments, that is, text preceded by a semicolon (;) not be included in the comparison. Otherwise, comments are compared like all other text.

The FORM_FEEDS argument specifies that lines beginning with a single form-feed character (CTRL/L) are to be excluded from the comparison. The default is to include such lines in the comparison.

The SPACING argument specifies that any sequence of blank and tab characters is to be interpreted as a single blank for the comparison. This argument is useful when comparing source files that differ only in their spacing. The output listing includes blanks and tabs as they are found in the input files, but the blanks and tabs are ignored. The default is to compare sequences of blanks and tabs like any other characters.

The TRAILING_BLANKS argument specifies that all blanks following the last nonblank character in the line be ignored in the comparison. The default is to compare trailing blanks like all other characters. If you specify both TRAILING_BLANKS and COMMENTS in your /IGNORE qualifier, any blanks preceding a semicolon (;) are also ignored.

The DCL DIFFERENCES command does not have quite the same defaults as CMP, the File Compare Utility, which DIFFERENCES invokes. Specify /IGNORE:(BLANK_LINES,FORM_FEEDS) to duplicate the CMP defaults for blank lines and form feeds. Other CMP defaults need not be specified.

```

/LINES:n

```

Specifies that n lines must be identical before a match is recognized. The default value for n is 3 identical lines.

DIFFERENCES (Cont.)**/[NO]NUMBERS**

Specifies whether or not lines in output file are preceded by line numbers. /NONUMBERS specifies that lines in the output file not be preceded by line numbers. The standard output listing automatically includes line numbers. Line numbers start with 1 in increments of one. All lines are numbered, including blanks.

/OUTPUT:filespec

Specifies that the output listing be written to the file named in the qualifier. Normally, this output appears on your terminal.

/SLP[:au]

Specifies that the output listing take the form of a SLP indirect command file. When you include this qualifier, the output listing is an SLP indirect command file that makes infile1 identical to infile2. The optional au argument, preceded by a colon, is a 1- through 8-character alphanumeric audit trail symbol. For more information on the Source Language Input Program (SLP), see the RSX-11M/M-PLUS Utilities Manual.

Examples

The following examples use several slightly different versions of the same MACRO-11 source file for comparisons. The full text of version 1 is printed here. Version 2 is identical except for that one Executive directive is different; version 3 has additional blank lines inserted. In each instance, the "IDENT" number matches the version number.

Here is the text of version 1:

```
.TITLE WRITE PRINT A PREDEFINED MESSAGE ON THE USER TERMINAL
.IDENT "01"

.MCALL ALUN$C, QIOW$C, EXST$C

STRING: .ASCII "This is a simple MACRO-11 program."
LENGTH =    .-STRING          ; CALCULATE THE STRING LENGTH
.EVEN

START: ALUN$C 1,TI,0           ; ASSIGN THE LUN TO USE TO THE TERMINAL
      QIOW$C IO.WVB,1,1,,,,STRING,LENGTH,40>
      ; PRINT THE STRING
      EXST$C EX$SUC           ; SUCCESSFUL EXIT IN ALL CASES

.END START                    ; SETUP THE STARTING ADDRESS
```

In the following examples, a "difference" as defined by the DIFFERENCES command consists of one or more lines that differ in the two files being compared. Each "difference" ends when the first matching line is encountered. The listing of the "difference" includes all the nonmatching lines AND the first matching line.

DIFFERENCES (Cont.)

```

$ DIFFERENCES/IGNORE:(BLANK_LINES,FORM_FEEDS) (RET)
Filespec1? TOM.MAC;1 (RET)
Filespec2? TOM.MAC;2 (RET)
*****
1) DB2:[303,5]TOM.MAC;1
2      .IDENT  "01"
4      .MCALL  ALUN$C, QIOW$C, EXST$C
6  STRING: .ASCII "This is a simple MACRO-11 program."
*****
2) DB2:[303,5]TOM.MAC;2
2      .IDENT  "02"
4      .MCALL  ALUN$C, QIOW$C, EXST$S
6  STRING: .ASCII "This is a simple MACRO-11 program."
1 differences found
TI:/-BL/-FF/-CB=TOM.MAC;1,TOM.MAC;2

```

In this example, the user compared two files that differ only in one Executive directive -- EXST\$S or EXST\$C -- and in their version numbers. The user specified the following:

```
/IGNORE:(BLANK_LINES,FORM_FEEDS)
```

to reproduce the defaults of CMP, the File Compare utility, which the DIFFERENCES command invokes.

The listing of differences between the two files starts with the IDENT numbers on line 2 of the files. The next line is blank in both files. Although the line is numbered, it is not printed on the terminal and is ignored in the comparison. It would not change the results if one file had the blank line there and one did not.

The next listed difference is in the Executive calls on line 4. Again, the blank lines are not shown. The next line of code listed is the same in both files. This marks the end of a group of lines that are different in the two files.

The command line reprinted at the bottom of the listing is the CMP command line equivalent to the DIFFERENCES command.

```

$ DIFFERENCES TOM.MAC;1 TOM.MAC;2 (RET)
*****
1) DB2:[303,5]TOM.MAC;1
2      .IDENT  "01"
3
4      .MCALL  ALUN$C, QIOW$C, EXST$C
5
*****
2) DB2:[303,5]TOM.MAC;2
2      .IDENT  "02"
3
4      .MCALL  ALUN$C, QIOW$C, EXST$S
5

1 differences found
TI:/BL/FF/-CB=TOM.MAC;1,TOM.MAC;2

```

DIFFERENCES (Cont.)

This example illustrates the defaults for the DIFFERENCES command. The results are the same, but the blank lines are printed. If one file had had more blank lines than the other, the results would not have been the same. See the examples comparing version 1 and version 3.

```
$ DIFFERENCES/CHANGE_BAR/IGNORE:(BLANK LINES,FORM FEEDS) TOM.MAC;1 TOM.MAC;2 (RET)
 1 .TITLE WRITE PRINT A PREDEFINED MESSAGE ON THE USER TERMINAL
 2 ! .IDENT "02"
 4 ! .MCALL ALUN$C, QIOW$C, EXST$S
 6 ! STRING: .ASCII "This is a simple MACRO-11 program."
 7 LENGTH = .-STRING ; CALCULATE THE STRING LENGTH
 8 .EVEN
10 START: ALUN$C 1,TI,0 ; ASSIGN THE LUN TO USE THE TERMINAL
11 QIOW$C IO.WVB,1,1,,,STRING,LENGTH,40>
12 ; PRINT THE STRING
13 EXST$C EX$SUC ; SUCCESSFUL EXIT IN ALL CASES
15 .END START ; SETUP THE STARTING ADDRESS
 1 differences found
TI:/-BL/-FF/CB/VB::041=TOM.MAC;1,TOM.MAC;2
```

This example shows the effect of the /CHANGE_BAR qualifier. The second file is printed on the terminal with exclamation points next to the numbers of the changed lines.

```
$ DIFFERENCES/IGNORE:(BLANK LINES) TOM.MAC;1 TOM.MAC;3 (RET)
*****
1) DB2:[303,5]TOM.MAC;1
 2 .IDENT "01"
 4 .MCALL ALUN$C, QIOW$C, EXST$C
*****
2) DB2:[303,5]TOM.MAC;3
 2 .IDENT "03"
 4 .MCALL ALUN$C, QIOW$C, EXST$C

 1 differences found
TI:/-BL/FF/-CB=TOM.MAC;1,TOM.MAC;3
```

Versions 1 and 3 differ only in their blank lines and version numbers. In this example, with /IGNORE:(BLANK LINES), the only difference is in the IDENT numbers. The listing of the difference ends with the first line that is not different. Blank lines are numbered but otherwise ignored.

DIFFERENCES (Cont.)

```

$ DIFFERENCE TOM.MAC;1 TOM.MAC;3 (RET)
*****
1) DB2:[303,5]TOM.MAC;1
2      .IDENT "01"
3
*****
2) DB2:[303,5]TOM.MAC;3
2      .IDENT "03"
3
*****
1) DB2:[303,5]TOM.MAC;1
7      LENGTH =      .-STRING      ; CALCULATE THE STRING LENGTH
*****
2) DB2:[303,5]TOM.MAC;3
7
8      LENGTH =      .-STRING      ; CALCULATE THE STRING LENGTH
*****
1) DB2:[303,5]TOM.MAC;1
11     QIOW$C10.WVB,1,1,,, <STRING,LENGTH,40>
*****
2) DB2:[303,5]TOM.MAC;3
12
13     QIOW$C10.WVB,1,1,,, <STRING,LENGTH,40>

3 differences found
TI:/BL/FF/-CB=TOM.MAC;1,TOM.MAC;3

```

Comparing the same two files without specifying /IGNORE:(BLANK_LINES) has quite different results. Three differences are found. The first is the differing IDENT numbers. The other two differences are nothing more than blank lines.

Notes

You can compare any two text files. They need not be MACRO-11 source files, but most qualifiers to DIFFERENCE are for use with MACRO-11 files.

The DIFFERENCES command invokes CMP, the File Compare Utility. See the RSX-11M/M-PLUS Utilities Manual.

UNLOCK

4.4.5 UNLOCK

UNLOCK unlocks 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[s]]
File(s)? filespec[s]
```

```
UNLOCK[/qualifier[s]] filespec[s]
```

Command Qualifiers

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

Parameter

filespec

Identifies the locked file(s) that you wish to unlock.

Command Qualifiers

```
/DATE:dd-mmm-yy
```

Specifies that you wish the UNLOCK command to affect only files created on the date given.

```
/SINCE:dd-mmm-yy
```

Specifies that you wish the UNLOCK command to affect only files created on or since the date given.

```
/THROUGH:dd-mmm-yy
```

Specifies that you wish the UNLOCK command to affect only files created on or before the date given.

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

Specifies that you wish the UNLOCK command to affect only files created within that range.

```
/TODAY
```

Specifies that you wish the UNLOCK command to affect only files created on the same day the command is being issued.

UNLOCK (Cont.)

/EXCLUDE:filespec

Specifies that you do not want the UNLOCK command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

Example

```
$ Directory TOPEKA.KAN (RET)
Directory DB0:[303,5]
2-MAY-85 13:13

TOPEKA.KAN;3    32.    L    25-APR-85 12:29

Total of 32./40. blocks in 1. files
$
$
$ UNLOCK TOPEKA.KAN (RET)
$
$
$ DIRECTORY TOPEKA.KAN (RET)
Directory DB0:[303,5]
2-MAY-85 13:14

TOPEKA.KAN;3    32.          25-APR-85 12:29

Total of 32./40. blocks in 1. files
```

In this example, the user found a locked file, perhaps through an error message stating that the file could not be opened. The locked state is indicated by the L in the third column of the directory listing. After the user issues the UNLOCK command, the file is no longer locked.

Notes

In many cases, a locked file has no contents because the task that opened the file aborted before writing to the file. In this case, you do not need to unlock the file before deleting it.

Another common case of file locking is a data file that has been opened by some task that aborted or exited without closing the file. In this case, you have to unlock the data file before running the task again. If the file is written to by the task, it may have been corrupted.

4.5 SET AND SHOW

You can SET and SHOW your default device and UFD. You can SET PROTECTION on any file in your directory or on a volume mounted on your private (allocated) device.

SET DEFAULT

4.5.1 SET DEFAULT

SET DEFAULT establishes your default device or directory, or both.

Format

```
SET DEFAULT [ddnn:][[g,m]]
```

Parameters

If you issue SET DEFAULT without parameters, your default device and directory return to your login defaults, if you are nonprivileged. Privileged users must explicitly reset their login directory.

ddnn:

Specifies the name of the device you wish to be your default device. The device must be a Files-11 device. The device name can be a logical name, but the SHOW DEFAULT display names the physical device.

[g,m]

Specifies the directory you wish to be your default directory.

When you change your directory, the system checks your login UIC for your privilege. If you are nonprivileged, you remain nonprivileged even if your new default directory has a group number of 10 or less.

If you are privileged, this command changes your login UIC as well as your default directory, but your privileges remain the same. In some cases, however, privileged users may have to reset their directory to a privileged group number in order to use privileged commands.

Examples

```
$ SET DEFAULT DR1:[303,5] (RET)
```

This example sets DR1: as the default device and [303,5] as the default directory. All subsequent commands default to this device and directory unless you explicitly state otherwise.

```
$ SHOW DEFAULT (RET)
DB1:[303,5] TT56:
$ SET DEFAULT DB2:[200,200] (RET)
$ SHO DEF (RET)
DB2:[200,200]
$ SET DEF (RET)
$ SHO DEF (RET)
DB1:[303,5]
```

SET DEFAULT (Cont.)

This example illustrates the use of the unadorned SET DEFAULT command. The user started with the login default device of DB1: and the directory [303,5], as given in the display from the SHOW DEFAULT command. Then, the user set the defaults to DB2:[200,200] and checked the setting with SHOW DEFAULT. Finally, the user issued SET DEFAULT without specifying any parameter. The user's defaults are returned to the login defaults. See notes.

```
$ SET DEF DK1: (RET)
```

```
$ ASSIGN DK1: SY0: (RET)
```

The two commands in this example are equivalent. In each case, the default device becomes DK1:. When you set the default device, SET DEFAULT actually assigns SY0: as a logical name for the device you wish. SY0: is the logical device name that represents your default device.

Notes

SET DEFAULT is an unusual command in that it has differing effects depending on the presence or absence of its parameters.

The only way to change your default directory in DCL is through SET DEFAULT. When you issue SET DEFAULT including a directory, DCL directly alters the default directory held in your Unit Control Block (UCB).

When you change your default device, however, SET DEFAULT generates the equivalent of an assignment of the logical name SY0: to the physical device you name, as shown in the examples.

Conversely, when you issue the command without any parameters, SET DEFAULT generates the equivalent of a deassignment of the logical name SY0:, meaning that your default device assignment returns to your login SY0:, which is whatever device you log in to, and, for nonprivileged users resets your directory to your login directory.

Because DCL is issuing two commands, it is possible that one may succeed while the other fails.

SHOW DEFAULT

4.5.2 SHOW DEFAULT

SHOW DEFAULT displays the current default device and directory for your terminal.

Format

```
SHOW DEFAULT
```

Examples

```
$ SHOW DEFAULT (RET)
DB1:[303,5] TT56:
```

This example shows the display from SHOW DEFAULT, giving the default device and directory, plus the terminal name.

```
$ SHOW DEFAULT (RET)
DB1:[303,5] TT56:
$ SET DEFAULT DB0: (RET)
$ SHOW DEF (RET)
DB0:[303,5] TT56:
$ SET DEFAULT DK1:[1,7] (RET)
$ SHOW DEF (RET)
DK1:[1,7] TT56:
$ ASSIGN DT2: YY1: (RET)
$ SET DEF YY1: (RET)
$ \SHOW DEF (RET)
DT2:[1,7] TT56:
```

In this example, the user first checked the defaults with SHOW DEFAULT, and then used SET DEFAULT to change the default device from DB1: to DB0:.. The user checked with SHOW DEFAULT and then set the defaults to device DK1: and directory [1,7]. After checking again, the user assigned the logical name YY1: to DT2: and set YY1: as the default device. Note that the subsequent SHOW DEFAULT display did not give the logical name YY1:, but the physical name DT2:..

Notes

See Chapter 5 for more information on logical assignments.

SET PROTECTION**4.5.3 SET PROTECTION**

SET PROTECTION establishes the protection status of files.

Protection status determines which categories of user may access a file and what each kind of user may do to the file.

Nonprivileged users can set the protection of any files in their own directory.

Privileged users can set the protection of any file.

Format

```
SET PROTECTION:code[/qualifier[s]] filespec[s]
```

```
SET PROTECTION[/qualifier[s]]
File? filespec[s]
Code? (code)
```

```
SET PROTECTION[/qualifier[s]] filespec[s] (code)
```

Command Qualifiers

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

Parameters

filespec[s]

Specifies the file or files for which protection is to be established. Multiple file specifications, separated by commas, and wildcards are permitted. All files you specify receive the protection status you specify.

code or (code)

Specifies which classes of users are permitted access to the file and what each class of user may do to the file. The parentheses are required syntax.

There are four kinds of users:

SYSTEM	The operating system itself, and privileged users, those with group numbers of 10 or less.
OWNER	The user having the same UIC as that the file was created under.
GROUP	All users with the same group number as that the file was created under.
WORLD	All other users.

SET PROTECTION (Cont.)

There are also four kinds of access to files:

READ	The user, or the user's tasks, may read, copy, print, type the file, or run it, if it is a task image.
WRITE	The user, or the user's tasks, may add new data to the file by writing to it.
EXTEND	The user, or the user's tasks, may increase the amount of disk space allocated to the file.
DELETE	The user, or the user's tasks, may delete the file.

The access you specify is the access that is given. If you specify only READ access to users in a particular category, those users will be able to type, print, copy, or run (if it is task image) your files. However, your files are protected from being edited or deleted by those users.

The system default protection code is expressed as follows:

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

Under this code, the system, and privileged users, have full access to your files. You, as well as others with your group number, also have full access to your files. Other nonprivileged users can only read your files.

All files have system default protection unless you change their protection codes using the SET PROTECTION command. In addition to this protection, there is default volume protection. See Section 4.1.6.

If you wish to protect a file differently, use the same format, naming only the user group whose access rights you wish to change and the access form you wish to grant to that group. If you wish to deny all access to a group, simply name the group and omit the colon (:) and the code for the access form.

Whenever any command specifies the protection code, enter it in this form.

Command Qualifiers

/DATE:dd-mmm-yy

Specifies that you wish to set the protection only on files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you wish to set the protection only on files created on or since the date given.

SET PROTECTION (Cont.)

/THROUGH:dd-mmm-yy

Specifies that you wish to set the protection only on files created on or before the date given.

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

Specifies that you wish to set the protection only on files created within that range.

/TODAY

Specifies that you wish to set the protection only on files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you do not want the SET PROTECTION command to affect certain files. The filespec argument to /EXCLUDE can contain either a single file specification (for example, TEST.DAT;2), or wildcards in all but one of the file specification fields (for example, *.DAT;*; TEST.*;*; or *.*;2) if you want to exclude multiple files. However, the version number must be given, either explicitly or as a wildcard (*).

Examples

```
$ SET (RET)
FUNCTION? PROTECTION (RET)
CODE? (SYSTEM:RWED,OWNER:RWED,GROUP,WORLD) (RET)
$ File(s)? SANTONE.TEX (RET)
$
$ DIRECTORY/FULL SANTONE.TEX (RET)
Directory DB1:[303,5]
5-MAR-85      10:49

SANTONE.TEX;1      (10732,11)      2./5.      4-MAR-85 9:49
[303,5]      [RWED,RWED,,]

Total of 2./5. blocks in 1. file
```

This example sets file protection so that privileged users and the owner have full access to the file and all others have no access to the file. The DIRECTORY/FULL command displays the protection of files. The list is in the same order as before: SYSTEM, OWNER, GROUP, WORLD. The commas separate the access rights of each group. The commas with nothing following them signify no access for other users with the same group number and no access for WORLD.

```
$ SET PROTECTION *.*;* (SYS:RWE,OWN:RWE,GRO:RW,WOR:RW) (RET)
```

This example sets the protection for all files in the current default directory. No class of user can delete the files in the directory; other nonprivileged users cannot extend files in the directory.

SET PROTECTION (Cont.)

Notes

Although you can protect your files against access by privileged users, privileged users can change the protection status of any file. Thus, you can prevent privileged users or the system from inadvertently accessing your files, but you cannot prevent them from changing the protection status and doing as they want.

Other commands accepting a protection code include INITIALIZE, INITIALIZE/UPDATE, MOUNT, and CREATE/DIRECTORY.

If you deny READ access to a task image file, the task cannot be run.

SET FILE

4.5.4 SET FILE

SET FILE establishes certain file attributes. You can change an end-of-file marker, have an entry in one directory point to a file in another directory, remove an entry from a directory, or truncate files to their actual length.

Format

```
SET FILE[/qualifier[s]] filespec[s][/qualifier[s]]
```

Command Qualifiers

File Qualifiers

```
/ENTER:synonym_filespec
```

```
/END_OF_FILE:(BLOCK:n, BYTE:n)
```

```
/NOWARNINGS
```

```
/REMOVE
```

```
/REWIND
```

```
/TRUNCATE
```

Parameter

```
filespec[s]
```

Identifies the file or files that you want the operation to affect.

Command Qualifiers

```
/ENTER:synonym_filespec
```

Use this qualifier if you want to be able to refer to a file by more than one name, or if you want to have the same file in more than one directory. The argument `synonym_filespec` is the new directory entry. If many users need the same file, this qualifier can save you a lot of disk space.

If there is already a file of the name you specify, the new file will be one version number higher than the existing file.

If you specify more than one file with this qualifier, the `synonym_filespec` argument points to all those files. In effect, the files are concatenated.

If you omit any field of the `synonym_filespec`, the corresponding field of the parameter file specification is used in its place. The default parameter file specification is `*.*;*`.

/NOWARNINGS

Suppresses error messages from the operation.

/REMOVE

Removes a directory entry from a directory. The `/REMOVE` qualifier is the opposite of the `/ENTER` qualifier.

You can use this qualifier to delete synonyms created by the `/ENTER` qualifier. You can also use it to eliminate directory entries which, for whatever reason, point to nonexistent files. If you remove the last entry for a file, it can be found only with the Verify Utility (VFY). Be careful.

SET FILE (Cont.)**/REWIND**

Causes the tape to rewind before the operation begins. This qualifier is to be used with magnetic tapes only.

/TRUNCATE

Enables you to reclaim unused disk space. When the RSX-11M system creates a file, it allocates file space in multiple units, usually five blocks at a time. The system, however, uses these blocks only one at a time. This means that there are often unused blocks included in the files.

Issue the DIRECTORY/FULL command to check the number of blocks used and allocated for a file. Two numbers in the form 23./25. in the listing indicate that 23 blocks are used out of 25 allocated. Because EXTEND access allows the system to change the file size whenever it wants to, the extra blocks may be of only limited use and can be truncated without loss.

In some cases, the presence of extra allocated but unused blocks may speed up the execution of the task.

File Qualifiers**/END_OF_FILE: [(BLOCK:n, BYTE:n)]**

You can specify only one file with this qualifier. Use /END_OF_FILE to specify where a file's end-of-file pointer is. This helps in situations, such as system crashes, when a file contains useful information, but its end-of-file pointer is wrong, preventing you from obtaining the information. The end-of-file pointer is an unprotected file attribute. If you are the file owner or a privileged user, you do not need READ or WRITE access to read or change this attribute. If you are classified as group or world relative to the file, you need READ access to read the attribute and WRITE access to change it.

If you do not specify values for BLOCK and BYTE, the system puts the end-of-file pointer past the last byte of the last block allocated to the file.

For BLOCK, the argument n specifies the block number where the end-of-file pointer is to be placed. Usually, the pointer cannot be placed beyond the highest number of blocks allocated to the file. However, if all the bytes of the allocated blocks are used, then the pointer can be placed in the first byte of the next block. For example, if the number of blocks allocated is 22, you would specify (BLOCK:23, BYTE:0). The block number can be either octal (n) or decimal (n.).

For BYTE, the argument n specifies the location of the first unused byte of the specified block. The byte number can be octal (n) or decimal (n.). The maximum value for BYTE is 777(octal), or 511(decimal).

SET FILE (Cont.)

Examples

```

$ DIRECTORY/FULL HAI.FAI (RET)
DIRECTORY DU1:[222,314]
16-APR-85 11:02
HAI.FAI;1      (10,10)      0./55.
16-APR-85 11:02
[222,314][RWED,RWED,RWED,R]
Total of 0./55. blocks in 1. files
$ SET FILE HAI.FAI/END_OF_FILE:(BLOCK:56,BYTE:0) (RET)

```

In this example, the user has just aborted a task that writes data in the file HAI.FAI. On checking the directory for HAI.FAI with the /FULL qualifer, the user finds that while 55 blocks are allocated, none have been used. The file may contain nothing or meaningless information. But it may contain useful data. In any case, the end-of-file pointer can be moved out to just past the last allocated block. The user issues the SET FILE /END_OF_FILE command and specifies byte 0 of the first unused block (55 + 1) as the end of the file. This makes it possible to examine the contents of the 55 allocated blocks.

```

$ SET FILE/ENTER:DR0:[301,7]LANE.TXT ODIN.TXT (RET)

```

In this example, the user caused the file ODIN.TXT to appear in the [301,7] directory under the name LANE.TXT. There is still only one copy of the file, and only a small amount of extra disk space is used.

```

$ SET FILE/REMOVE [301,7]LANE.TXT (RET)

```

In this example, the user decided that the directory [301,7] is no longer needed to access the file LANE.TXT, and removed the entry from the directory.

```

$ DIRECTORY/SUMMARY (RET)
Total of 146./155. blocks in 5. files
$ SET FILE *.*;*/TRUNCATE (RET)
$ DIRECTORY/SUMMARY (RET)
Total of 146./146. blocks in 5. files

```

In this example, the user uses the DIRECTORY/SUMMARY command to determine that there are nine unused blocks in the directory. Because blocks allocated cannot be used, the user issues the SET FILE /TRUNCATE command with wildcards specifying that all files in the directory are to be truncated. Now there are no unused blocks in the directory.

CHAPTER 5

HANDLING DEVICES AND VOLUMES

Peripheral devices are the actual hardware elements that, with the computer, make up a computer system. In general, a peripheral device is anything that is not part of the CPU or main memory.

Peripheral devices handle all input and output from the system. One primary function of the operating system is to manage efficiently all the peripheral devices in the system.

System generation informs the Executive which devices are to be included in your installation and describes the characteristics of those devices.

In RSX-11M terminology, a file is an owner-named area on a volume. A volume is a collection of files or data stored on a magnetic medium, such as a disk or magnetic tape. This medium must be physically placed on a drive and logically mounted before the system can access the data on the volume.

This chapter briefly introduces the important concepts concerning devices and volumes. In addition, the chapter describes the DCL commands that deal with devices and volumes.

5.1 DEVICES ON RSX-11M SYSTEMS

Table 5-1 lists the devices found on RSX-11M systems. Each installation has a different mixture of physical devices. Each physical device has an associated hardware controller, which serves as an interface between the device hardware and the CPU. In addition, each device has a device driver, which is the software interface between the operating system and the device controller.

The table also lists pseudo devices found on RSX-11M systems. A pseudo device is an entity the system or user treats as an I/O device, although it is not actually any particular physical device. A pseudo device name is a pseudonym through which actual physical devices can always be reached. See Table 5-2 and Section 5.1.2.1 for more information on pseudo devices.

Individual devices are identified by a 2-letter mnemonic and an octal unit number, terminated by a colon. If you omit the unit number, the system defaults to unit number 0. That is, if you wish to specify DB0: in a command, DB: will do.

HANDLING DEVICES AND VOLUMES

Table 5-1
Devices on RSX-11M Systems

Mnemonic	Device
AD:	Analog-to-Digital Converter
AF:	Analog-to-Digital Converter
AR:	Laboratory Peripheral Subsystem
CL:	Console Listing Pseudo Device
CO:	Console Output Pseudo Device
CR:	Card Reader
CT:	Cassette (TU60)
DB:	RP04/RP05/RP06 disk drives
DD:	DECTape II (TU58)
DK:	RK05/RK05J/RK05F disk drive
DL:	RL01/RL02 disk drives
DM:	RK06/RK07 disk drives
DP:	RP02/RP03/RPR02 disk drives
DR:	RM02/RM03/RM05 disk drives and RM80 fixed disk drive
DS:	RS03/RS04 fixed disk drives
DT:	DECTape (TU56)
DU:	RA60/RA80/RA81 disk drives RC25 disk drive RD51/RD52/RD53 disk drives RX50 diskette drive
DX:	RX01 diskette drive
DY:	RX02 diskette drive
EM:	Non-Rotating Electronic Memory (ML11)
GR:	Graphics Display Processor and Scope
IC:	Industrial Control System
IS:	Industrial Control Subsystem
LA:	Laboratory Peripheral Accelerator
LB:	System Default Pseudo Device
LP:	Line Printer
LS:	Laboratory Subsystem
MF:	TU78 magnetic tape drive
MM:	TU45/TU16/TE16/TU77 magnetic tape drives
MS:	TK25/TS11/TS05/TU80 magnetic tape drives
MT:	TS03/TU10/TE10 magnetic tape drives
MU:	TK50 magnetic tape drive
NL:	Null Device
PP:	Paper Tape Punch
PR:	Paper Tape Reader
SY:	User Default Pseudo Device
TI:	Terminal Input Pseudo Device
TT:	Terminal
UD:	Universal Digital Controller
WK:	Workfile Pseudo Device
XB:	DA11-B Interprocessor Link
XE:	DEUNA Ethernet Interprocessor Link
XL:	DL11-E Interprocessor Link
XM:	DMC-11 Interprocessor Link
XP:	DP11 Synchronous Interface
XQ:	DQ11 Synchronous Interface
XU:	DU11 Synchronous Interface
XW:	DUP11 Synchronous Interface

HANDLING DEVICES AND VOLUMES

RSX-11M systems support a variety of devices, including the terminal, line printer, disk, and tape unit. The line printer is called a unit record device. Disks and magnetic tapes are mass-storage devices. Disks are random-access devices and magnetic tapes are sequential-access devices. Each of these device types is discussed briefly in this section and, when necessary, in more detail elsewhere in this chapter.

Devices are informally identified by the 2-letter mnemonic identifying the device driver (such as DL: for RL02 disk), but you should remember that the driver may support more than one device. For instance, the DU: driver supports both RD51 and RD52 fixed disks, and the RX50 diskettes.

Unit record devices operate on data one record at a time. Unit record devices can store only one record's worth of data at a time. For instance, a line printer (LP:) receives and stores characters until it has a full line and then prints that line. Unit record devices on RSX-11M include the paper tape punch (PP:), paper tape reader (PR:), the TU60 cassette (CT:), and card reader (CR:).

For storing large amounts of data, disks and tapes are used. These disks and tapes are called magnetic media. The information stored on these media must be arranged in formats suitable to the operating system and your installation's software.

On RSX-11M systems, the information contained on the magnetic medium is called a volume. Volumes can be of several varieties. Foreign volumes are those on which the data is organized along principles not recognized by RSX-11M systems. Files-11 volumes are those on which the data is organized according to the normal file structure used by RSX-11M systems. Devices that accept media containing Files-11 volumes are all disk devices, including diskettes and disk packs (DB:, DF:, DK:, DL:, DP:, DR:, DS:, DU:, DX:, and DY:), the TU56 DECTape I (DT:), and the TU58 DECTape II (DD:). Files-11 volumes are created through the INITIALIZE command (Section 5.9). The operating system treats all Files-11 volumes identically, regardless of the device or medium.

NOTE

DECTapes are random-access devices. Therefore, in this manual and other system documentation, the term "disk" includes DECTapes and the term "magnetic tape" excludes DECTapes.

Files-11 volumes must be mounted (using the MOUNT command, Section 5.7) for the system to access them. Mounting a volume brings in a task called an Ancillary Control Processor (ACP), which serves as an interface between system and user tasks and the device driver. The ACP also maintains the Files-11 structure on the disk. The ACP that controls these functions is called F11ACP.

Magnetic tapes are a sequential-access medium. Random-access media, such as disks, are used where speed is the most important virtue. Magnetic tapes are used where economy and transportability are more important than speed.

There are two kinds of magnetic tape volumes: labeled and unlabeled. Labeled magnetic tape volumes are accessed through the magnetic tape ACP (MTAACP) and conform to the American National Standard for Magnetic Tape Labels and File Structure for Information Exchange (ANSI X3.27-1978). You can use unlabeled tapes for backup on RSX-11M

HANDLING DEVICES AND VOLUMES

systems. These tapes, which are in a nonstandard format, must be mounted using the /NOLABEL qualifier. You can use labeled tapes (the default) for operations that require ANSI format (for example, COPY, DELETE, and DIRECTORY).

Laboratory and industrial devices are common in process-control environments. These devices include the following:

- Analog-to-digital converters (AD:, AF:)
- Laboratory peripheral subsystem (AR:)
- Graphics display processor (GR:)
- Industrial control systems (IC:, IS:)
- Laboratory devices (LA:, LS:)
- Universal digital controller (UD:)

The system uses several physical devices in its operations. These devices include:

- Communications devices, including parallel communications links (LT:, LR:)
- Interprocessor links (XB:, XE:, XL:, XM:)
- Synchronous interfaces (XP:, XQ:, XU:, XW:)

There are system control peripherals not normally accessed by users (BS:, MK:, II:).

Finally, there are the pseudo devices, a general term used to identify a number of system features implemented either through device drivers without associated hardware or through logical device names. The pseudo devices include CO:, CL:, LB:, NL:, RD:, SY:, TI:, VT:, and WK:. See Section 5.1.2.1.

5.1.1 Volumes and Devices

Because the information in the volume must be on some medium and because that medium must be physically loaded on a device drive, the device driver mnemonic sometimes denotes the device itself and sometimes it denotes the volume. For instance, if you have files on a volume to which you wish to have private access, you take the following steps:

1. Check with SHOW DEVICES dd: to see which devices of the type you wish to use are available. The type you wish to use is determined by the magnetic medium the volume is on.
2. Make the device your private device through the ALLOCATE command.
3. Physically place the magnetic medium on the device drive. (This may be done for you by your system operator.)
4. Mount the volume through the MOUNT command.

If the device you choose is DUL:, both the ALLOCATE and the MOUNT command refer to DUL:. In the ALLOCATE command, DUL: is the device itself, while in the MOUNT command, DUL: is the volume. In most

HANDLING DEVICES AND VOLUMES

cases, this distinction is not important, but you should understand that the expression `ddnn:` refers sometimes to a physical device and sometimes to a volume on the device.

5.1.2 Physical Devices, Pseudo Devices, LUNs, and Logical Names

The purpose of devices of any kind is to handle I/O from tasks. RSX-11M systems offer you a number of ways to name and access physical devices. In addition to direct access to the physical devices by naming them in commands or file specifications, there are a number of forms of indirect access to physical devices.

The operating system often accesses physical devices as pseudo devices. Tasks access physical devices through logical unit numbers (LUNs) that establish a relationship between the I/O done by the task and the devices on the system. Finally, you can give any physical device a logical name.

5.1.2.1 Pseudo Devices

The names of pseudo devices are pseudonyms used by the system to access physical devices. System tasks are built to reference various pseudo devices. When the system is generated, the pseudo device names are associated with the physical devices that are actually present on the system. For instance, the system uses pseudo device `LB:` to refer to the disk location of the system libraries. At one installation, this might be `DR2:`, while at another, this might be `DB7:`, but regardless of the actual physical device involved, the name `LB:` will always refer to the current disk location of the system libraries.

Pseudo devices are implemented in several ways. Some pseudo device names are really only logical names. No system data structures are associated with logical names other than an entry in the system's Logical Unit Table (LUT). Other pseudo device names are implemented through redirection from one physical device to another. Redirection requires more system data structures: a Device Control Block (DCB) and a Unit Control Block (UCB). Finally, some pseudo device names are implemented through device drivers, which require not only full data structures but also the presence of the driver.

Note, however, the term pseudo device does not necessarily refer to a single form of software, but rather to the concept that a usable device name is known to the system regardless of the installation's device configuration.

Table 5-2 summarizes the pseudo devices on RSX-11M systems. Pseudo device `CO:` is handled differently depending on whether the system includes Console Logging. Without Console Logging, `CO:` is redirected, usually to the main operator's terminal, often `TT0:`. Including Console Logging means a driver called `CO:` is used. This driver permits your system manager to send system console output to any device or to a file. See the RSX-11M/M-PLUS System Management Guide for more information on Console Logging.

Pseudo device `CL:`, the Console Listing pseudo device, is commonly redirected to `LP0:`, the system line printer.

`LB:`, the Library pseudo device, is the disk on which the system directory is located.

HANDLING DEVICES AND VOLUMES

Table 5-2
Pseudo Devices on RSX-11M

Device Name	How Implemented	Comments
CO:	Redirection	No Console Logging
CO:	Driver	Console Logging present
CL:	Redirection	
LB:	Redirection	
NL:	Driver	
SY:	Assignment	User default device
TI:	Dynamic Redirection	
WK:	Assignment	
dd:	Driver	Physical devices, hardware required

Pseudo device NL:, the null device, is implemented through the NL: driver. Program testing often requires an available input or output device. The NL: pseudo device can be used for either of these purposes. For instance, if you are testing a program that normally generates large amounts of output, you can assign the appropriate LUN to NL: rather than wasting paper or disk space. The system discards output sent to NL:. On the other hand, if you are testing a program that ordinarily receives input from a device or file, NL: can generate usable input without any other source. When a task reads from NL:, the driver returns IE.EOF, the code for end of file. When a task writes to NL:, the driver returns IS.SUC, the code for success.

Pseudo device SY: can be created in two ways. First, SY: is always redirected to the device from which you bootstrap the system. Second, SY: is the logical name applied to the user's default or "home" device. This logical name is usually established by the LOGIN command and ACNT, the Account File Maintenance Program. This names the device to which all commands that do not specify a device name default. You can display your default device with the SHOW DEFAULT command. You can change your default device with the SET DEFAULT command or with the ASSIGN command. When the system determines which device to use, it looks for logical, pseudo, and physical device assignments, in that order.

Pseudo device TI: is always your terminal, regardless of its number or type. If you are logged in on two terminals, you have two TI:s. TI: is dynamically redirected on a per-command basis to your current terminal. TI: stands for terminal interface.

The WK: workfile pseudo device is used by a number of system tasks. WK: is named by assignment.

NOTE

Users can also assign logical names to devices (see the ASSIGN command, Section 5.3), redirect devices (see the ASSIGN/REDIRECT command, Section 5.3.1), and write drivers (see the RSX-11M Guide to Writing an I/O Driver.)

HANDLING DEVICES AND VOLUMES

5.1.2.2 LUNs

Each task includes logical unit numbers, or LUNs, which establish a relationship between the I/O done by the task and physical device units. This relationship can be different for each task.

Tasks may have from 0 through 250 LUNs. The maximum number of LUNs is set when the task is linked.

LUNs can be assigned or reassigned in three ways:

1. Any user can specify the ASG option at link time.
2. Privileged users can use ASSIGN/TASK (Section 5.3.2) to change the LUN assignments in an installed task.
3. The task can include the ALUN\$ Executive directive (or CALL ASNLUN from FORTRAN) to change LUN assignments during execution.

The Task Builder defaults build a task with six LUNs. The default assignments are as follows:

Device	LUN
SY0:	1.
SY0:	2.
SY0:	3.
SY0:	4.
TIO:	5.
CLO:	6.

This means a task linked with the default LUN assignments can simultaneously refer to up to as many as six files. These six files can be distributed as follows:

- Four files can be on SY:. (In this case, SY: is logical device name SY: as discussed in the previous section.)
- One file can be on pseudo device TI:, meaning that the terminal initiating the task can be used to communicate with the task.
- One file can be on pseudo device CL:, meaning that the task can send output to the console listing device, which is usually a line printer or hard-copy terminal.

Although the association of LUNs with physical or pseudo devices is established at link time, the actual assignment is not done until the task is installed, either explicitly or through the install-run-remove form of RUN.

Once the LUN is assigned, a task can map directly to a device without having to search device tables for the physical device. The LUN is simply a name used to represent the relationship between the logical device name and the physical device.

LUN assignments are established on a task-by-task basis, in contrast to assignments through the privileged ASSIGN/REDIRECT command, which apply to all tasks using a device.

HANDLING DEVICES AND VOLUMES

5.1.2.3 Logical Device Names

Logical device names provide a means by which tasks can maintain device independence. Logical device names have the same syntax as other device names, that is, `ddnn:`. The 2-character device name can be the same as a standard RSX-11M device or pseudo device, such as `DK:` or `LB:`, or it can use two letters with an arbitrary meaning, such as `AZ:`.

When you run a task that refers to a device by a logical name, first use the `ASSIGN` command to assign that logical name to the device. See Section 5.3.

There are three types of logical device assignment:

1. Local assignments, which can be made by any user, apply only to tasks initiated from the terminal used to make the assignment. Local assignments override both other categories of assignments.

Different users can assign the same local logical name to different devices.

2. Login assignments, which can be controlled only by privileged users through `ACNT`, the Account File Maintenance Program. Login assignments function as follows:

- They are reestablished each time the user logs in.
- They can also be established through `ASSIGN/LOGIN`, a privileged command.
- They override the next category, global assignments.
- They apply only to systems with multiuser protection.
- They remain in effect until the user logs off or until a privileged user deassigns them.

3. Global assignments, which can be made only by privileged users. Global assignments apply to all tasks running in the system.

The Executive keeps a table that records all logical assignments and their types. Each time a task assigns a LUN, the Executive refers to this table before checking the list of physical devices and pseudo devices. Thus, a logical name found in the table has precedence over a physical device unit having the same identifier.

5.1.3 Public, Shared, Private and Unowned Devices and Mounted Volumes

You must use the `MOUNT` command to access magnetic media. If a disk volume is in `Files-11` format, `MOUNT` grants access to the file system through the `Files-11` Ancillary Control Processor (`F11ACP`). If the information on the medium is in some other format, or is unformatted, `MOUNT/FOREIGN` bypasses the ACP and the file system. (`MOUNT/FOREIGN` is recommended for foreign volumes on RSX-11M. See Section 5.7). Use `MOUNT/FOREIGN` when initializing a volume or to use a custom ACP.

HANDLING DEVICES AND VOLUMES

Four categories of device ownership are supported on multiuser protection systems: Devices can be public, shareable, private, or unowned. Public devices are accessible to all. Within the limits of volume protection and file protection, shareable devices are accessible to all users who have mounted the device. Volumes on private devices are accessible only to the owner of the device. Volumes on unowned devices can be accessed on RSX-11M.

Volume protection and file protection can greatly alter your rights of access to a particular volume or file. See Section 4.1.6 for more information.

1. Public devices are accessible to all users; access is allowed automatically. Only a privileged user can set a device public. Devices are set public through the /PUBLIC qualifier to the SET DEVICE command (Section 5.11.2) or the /PUBLIC qualifier to the MOUNT command (Section 5.7). A public device cannot be mounted by any other user. The MOUNT/PUBLIC command sets the device public and deallocates it if it has been allocated. (An allocated device is a private device.)
2. Shareable devices are accessible to all users who mount the volume. A device is made shareable through the /SHAREABLE qualifier to the MOUNT command (Section 5.7). Nonprivileged users must know the volume label in order to mount the volume. Any number of users can mount the volume. It will remain mounted until the last user to mount the volume has dismounted it. You cannot mount a public or private device shareable.
3. Private devices cannot be accessed by anyone but the owner, that is, the person who allocated the device. Devices can be made private with the ALLOCATE command (Section 5.5), or by the /NOSHAREABLE qualifier to MOUNT, which also allocates the device. The DISMOUNT command will deallocate the device.

Users make devices private to prevent any other user from accessing the mounted volume. Any user can allocate a device, but your installation may have restrictions on the practice.

4. Unowned devices are those that have not been allocated, mounted, or set public. RSX-11M systems permit limited access to unowned volumes, but such use is not advised. See the discussion under the /FOREIGN qualifier to MOUNT (Section 5.7).

5.1.4 How to Prepare a Scratch Disk for Use

A scratch disk is a blank disk or a disk which contains data of no use. The prime example of a scratch disk is one that has never been used. Several system utilities and DCL commands are combined to prepare the scratch disk as a Files-11 volume. This section summarizes the process.

1. Allocate an appropriate device and place the disk on the drive. Make sure the device is write-enabled.
2. After physically placing the disk on its drive, spin it up and mount it using the /FOREIGN qualifier. This permits the system to access the disk but bypasses the file system. This step is optional.

HANDLING DEVICES AND VOLUMES

3. You may need to write sector and timing information about the physical blocks on the disk. In most cases, disk formatting is done by the manufacturer, but some disks are shipped unformatted. If you have any question about formatting a disk, see your system manager.

If formatting is necessary, you can use FMT, the Disk Volume Formatter Utility. (The RSX-11M/M-PLUS Utilities Manual provides full instruction on this utility.) To gain access to FMT from DCL, enter the following command:

```
$ RUN $FMT (RET)
FMT>
```

Now you can enter FMT commands.

4. Once the disk is ready, you must identify the bad blocks on the disk. These are blocks that are physically unusable for some reason. This information is often written on disks when they are manufactured, but you can supplement this information with the ANALYZE/MEDIA command.

```
$ ANALYZE/MEDIA ddnn:
```

5. Once you have a formatted disk with identified bad blocks, the disk driver and the disk controller have full use of the disk. In most cases, your scratch disk will already have formatting and bad-block data in place. Next, you must provide the information that the file system needs to use the disk.

This information is provided through the INITIALIZE command. Qualifiers to INITIALIZE permit you to set up your volume for the most efficient use by the system. The defaults for INITIALIZE produce a standard volume.

INITIALIZE writes a Volume Home Block, which contains information for the system about the initialization, including the volume label that is used to control access to the volume by nonprivileged users.

INITIALIZE also creates five files used by the system to locate or create other files on the volume. Here are brief descriptions of these files:

File	Description
000000.DIR	This is the Master File Directory (MFD). It is a file that 000000.DIR file contains a list of directories. The directories contain pointers to all named files on the volume. The MFD also contains the next four files.
INDEXF.SYS	This is the volume index file, which contains file-control information, such as file protection, creation date, the physical location of the file on the medium, and other information needed for the system to access the files on the volume.

HANDLING DEVICES AND VOLUMES

BADBLK.SYS This is the bad-block file. All unusable blocks are included in this file. Any block that is included in a file is unavailable for any other use.

BITMAP.SYS This file contains information on all blocks used and free on the volume. The information is in the form of one bit for every block. If a block is used, the corresponding bit is set.

CORIMG.SYS This is the operating system checkpoint file. It is included on every volume. If the volume is not to provide for checkpointing, the checkpoint file is zero blocks long.

6. Once the volume is initialized, it is no longer foreign. You must dismount it and mount it as a Files-11 volume. Use this sequence of commands:

```
$ DISMOUNT ddnn:  
$ MOUNT ddnn: volumelabel
```

Now the volume is a Files-11 volume. The file system has full access to the volume through a DIGITAL-supported ACP. If your system has a custom ACP for any device, you must mount the volume foreign and specify the ACP.

You can alter some of the data placed in the Volume Home Block by INITIALIZE by using the INITIALIZE/UPDATE command. See Section 5.9.1.

Certain system functions may also require that the volume be mounted foreign without specifying an ACP. You should check the appropriate documentation to be sure. This requirement may arise with the following DCL commands:

- BACKUP
- INITIALIZE

This requirement may also arise with the following utilities, described in the RSX-11M/M-PLUS Utilities Manual:

- FLX, File Transfer Utility Program
- FMT, Disk Volume Formatter Utility
- BAD, Bad Block Locator Utility
- BRU, Backup and Restore Utility, accessed from DCL through the BACKUP command
- DSC, Disk Save and Compress Utility Program
- VFY, File Structure Verification Utility
- DMP, File Dump Utility

HANDLING DEVICES AND VOLUMES

5.1.5 Preparing a Scratch ANSI Magnetic Tape for Use

This section describes how to prepare a scratch ANSI magnetic tape for use on RSX-11M. A scratch tape is a blank tape or a tape which contains data no longer of use to you. Follow these steps to prepare the tape for use with RSX-11M utilities and commands.

1. Allocate an appropriate device and place the tape on the drive. Make sure the device is write-enabled.
2. After you have physically placed the tape on the drive, mount the tape using the /FOREIGN qualifier. This permits the system to access the tape but bypasses the file system. This step is optional.
3. Next, you must provide information that the file system needs to use the tape. This is done through the INITIALIZE command. Qualifiers to INITIALIZE permit you to set up your volume for the most efficient use by the system. The defaults for INITIALIZE produce a standard ANSI magnetic tape volume.

INITIALIZE writes an ANSI volume label, which contains the File Set ID and also creates a dummy file. This dummy file is overwritten when the first user file is written to the tape.

There are no directories in the ANSI magnetic tape structure. See the RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual for more information on ANSI magnetic tapes.

4. Once the volume is initialized, it is no longer foreign. You must now dismount the volume and remount it as an ANSI volume, using DISMOUNT and then MOUNT (this time without the /FOREIGN qualifier).

5.1.6 Summary of Device and Volume Use Commands

All Files-11 volumes and ANSI and unlabeled magnetic tapes must be mounted for any access.

All other volumes are considered foreign volumes. Foreign volumes are normally mounted with the /FOREIGN qualifier to MOUNT.

On RSX-11M systems, you are not required to mount foreign volumes, but if you do not, the system and other users have unlimited access to the volume, as it is considered unowned. MOUNT/FOREIGN is recommended for most use of foreign volumes on RSX-11M systems.

Multiple users can mount a volume on a shareable device. The volume remains mounted until all users who have mounted it dismount it.

There are four kinds of device protection, based on device ownership:

1. A volume on a private (allocated) device can be mounted by the owner only.
2. A volume mounted /SHAREABLE by the first user to mount it can be mounted by any other user who knows the volume label. The volume stays mounted until the last user to mount it dismounts it.

HANDLING DEVICES AND VOLUMES

3. A volume on a public device is usually mounted after the device is set public. The system owns public devices. Mounting a public device means other users can access the device without mounting it. Only a privileged user can set a device public or nonpublic. Public devices cannot be allocated.
4. A volume on an unowned device can be mounted by anyone.

Any user who mounts a volume has full access to that volume within the limits of privilege and volume and file protection. Privileged users have privileged access. Nonprivileged users have nonprivileged access. Nonprivileged users have privileged access to volumes mounted on their private (allocated) devices.

A mounted device cannot be set public or nonpublic and cannot be allocated or deallocated.

LOGOUT issued by any user automatically dismounts any volumes mounted from that terminal and deallocates any devices allocated from that terminal. A mounted public device is not dismounted by LOGOUT, however.

ANALYZE/MEDIA

5.2 ANALYZE/MEDIA

ANALYZE/MEDIA allows you to identify and determine the number of bad blocks on a disk. ANALYZE/MEDIA determines if bad blocks exist on a disk volume and records their locations for use by the BACKUP and INITIALIZE commands.

Formats

```
ANALYZE/MEDIA [/qualifier[s]]
Device? ddnn:
```

```
ANALYZE/MEDIA [/qualifier[s]] ddnn:
```

Command Qualifiers

```
/ALLOCATE=label
/BADBLOCKS
/BADBLOCKS/NOEXERCISE
/[NO]EXERCISE[=(n,m)]
/OVERRIDE
/RETRY
/SHOW
```

Parameter

ddnn:

Specifies the device whose media you want to analyze.

Command Qualifiers

```
/ALLOCATE=label
```

Updates the bad block descriptor file and the file BADBLK.SYS in [0,0]. The volume must be mounted Files-11 to use this qualifier.

```
/BADBLOCKS
```

Prompts you for additional bad blocks, which can be entered in the bad block descriptor file along with the blocks already in the file.

```
/BADBLOCKS/NOEXERCISE
```

Updates the bad block descriptor file, but does not update the BADBLK.SYS file in [0,0]. The volume must be mounted Files-11 to use this qualifier.

```
/[NO]EXERCISE[=(n,m)]
```

Specifies the double word data pattern, (n,m), that ANALYZE/MEDIA uses to locate bad blocks. The parameter (n,m) represents the two 16-bit octal numbers used as the double word data pattern. You can specify a decimal number by placing a period after the number.

If you specify /NOEXERCISE, ANALYZE/MEDIA uses the blocks found in the bad block descriptor file without testing them.

ANALYZE/MEDIA (Cont.)**/OVERRIDE**

Ignores the last-track information on a last-track device and creates a bad block descriptor file on the last good block before the last track.

/RETRY

Specifies that the I/O driver retry the operation on bad blocks it finds before flagging them as bad.

/SHOW

Specifies that ANALYZE/MEDIA list bad blocks as it encounters them. The /SHOW qualifier does not list the bad blocks you enter manually using the /BADBLOCKS qualifier.

Examples

```
$ ANALYZE/MEDIA (RET)
Device? DUL: (RET)
```

In this example, ANALYZE/MEDIA analyzes the media found in DUL: and adds the bad blocks it encounters to the bad block descriptor file on the media. It does not display the bad block information on the terminal.

```
$ ANALYZE/MEDIA/BADBLOCKS DUL: (RET)
BAD>LBN(S)= 300 (RET)
BAD>LBN(S)= 1002 (RET)
BAD>LBN(S)= (CTRL/Z)
BAD -- DUL: Total bad blocks = 2.
```

In this example, ANALYZE/MEDIA prompts you for bad block information to add to the bad block descriptor file of DUL:. After you have entered all the bad blocks, press <CTRL/Z> to end the manual entry of bad blocks. ANALYZE/MEDIA then analyzes the entire volume and displays a message telling you the total number of bad blocks it finds, including those you entered manually. In this case, the only bad blocks on the media were those you entered. If you press <RET> instead of <CTRL/Z> when you finish entering bad block information, the command displays a list of the bad blocks you entered.

```
$ ANALYZE/MEDIA/BADBLOCKS/SHOW DUL: (RET)
BAD>LBN(S)= 3 (RET)
BAD>LBN(S)= 347 (RET)
BAD>LBN(S)= (CTRL/Z)
BAD -- DUL: Bad block found - LBN = 102.
                                     110.
```

In this example, ANALYZE/MEDIA found two bad blocks in addition to those you specified with the /BADBLOCKS qualifier. The /SHOW qualifier causes the command to display the additional bad blocks, but it does not display the blocks you entered.

ANALYZE/MEDIA (Cont.)**Note**

If your system includes error logging, the device you specify may have an error limit set. In this case, error logging displays the following message if the media reaches that limit:

```
ERRLOG -- **Warning: device ddnn: reached Hard limit (n.)
```

However, the ANALYZE/MEDIA command does not terminate at this point. It is an informational message only.

Error Messages

The ANALYZE/MEDIA command can display the following messages:

ANALYZE - Illegal analyze function

Explanation: You specified an incorrect syntax for the ANALYZE/MEDIA command.

User Action: Check the syntax and retry the command.

In addition, you may receive the following messages from BAD, the utility run by ANALYZE/MEDIA.

BAD - ddnn: Allocation Failure

Explanation: The ANALYZE/MEDIA command failed to allocate the block number sequence you entered. The I/O failed for a reason other than that the block number was already allocated to another file. This message applies to the /ALLOCATE qualifier only.

User Action: Either the volume is bad or the drive requires maintenance. Try another volume in the same drive.

BAD - ddnn: Bad block file not found

Explanation: The bad block descriptor file could not be read when you used the /BADBLOCKS/NOEXERCISE qualifier.

User Action: You must use the device without updating the bad block file, or reformat the device and destroy all data.

BAD - ddnn: Bad block file overflow

Explanation: The ANALYZE/MEDIA command detected more than 126(decimal) entries of bad blocks. This message usually indicates a device unit failure.

User Action: Either the volume is bad or the drive requires maintenance. Try another volume or contact your DIGITAL Field Service representative to fix the drive.

ANALYZE/MEDIA (Cont.)

BAD - ddnn: Bad block found - LBN = nnnnnn.

Explanation: Bad blocks are reported in this format, where LBN is the logical block number (decimal).

User Action: None. This message is informational and appears when you use the /SHOW qualifier.

BAD - ddnn: Block already allocated - LBN = numb

Explanation: The block number sequence you entered is already allocated to a file (the file may or may not be BADBLK.SYS). The value numb is the sequence you entered. The block sequence indicated by numb and the list of block numbers following numb were neither allocated to [0,0]BADBLK.SYS nor entered into the bad block descriptor file. This message applies only to the /BADBLOCKS qualifier.

User Action: Reenter the command line with another value.

BAD - ddnn: Block 0 bad - Do not use as system disk

Explanation: This is a warning message. When block zero is bad, a bootstrap block cannot be written on the disk; the disk is then useless as a system disk.

User Action: Label the disk to ensure that no one attempts to use it as a system disk.

BAD - Command I/O error

Explanation: The ANALYZE/MEDIA command did not recognize the command line entered from the keyboard.

User Action: Reenter the command line.

BAD - Command too long

Explanation: The command was longer than 80(decimal) characters.

User Action: Reenter the command line, using less than 80(decimal) characters.

BAD - Duplicate block number - numb

Explanation: The block number sequence you entered is already present in the bad block file. The value numb is the sequence you entered. BAD ignores any block number sequences you may have entered after the duplicate block numbers.

This message applies only to the /BADBLOCKS qualifier. If this message appears, it means that the block number that was allocated to [0,0]BADBLK.SYS already existed in the bad block descriptor file.

User Action: Reenter the command line with another value.

ANALYZE/MEDIA (Cont.)

BAD - ddnn: Failed to attach

Explanation: The ANALYZE/MEDIA command could not gain control of the device to be tested.

User Action: Determine if another task has attached the device. If so, wait until the task exits or abort the task to gain control of the device for BAD.

BAD - ddnn: Failed to read BADBLK.SYS header

Explanation: The message is self-explanatory and applies only to the /ALLOCATE switch.

User Action: The disk must be initialized using the DCL INI command.

BAD - ddnn: Failed to read Manufacturer's Bad Sector File

Explanation: A disk-read hardware error occurred while BAD was attempting to read the factory-written bad block data on the last-track device cartridge.

User Action: Reenter the command line with the /OVERRIDE switch included.

BAD - ddnn: Failed to read Software Bad Sector File

Explanation: The software-detected bad sector file could not be read in update mode.

User Action: Reenter the command line with the /OVERRIDE switch included.

BAD - ddnn: Failed to write Bad Block File

Explanation: The ANALYZE/MEDIA command could not write the bad block file. This condition usually results from a disk write error.

User Action: Reenter the command line. If the problem persists, the disk pack should be discarded.

BAD - ddnn: Fatal hardware error

Explanation: The message is self-explanatory.

User Action: Contact your DIGITAL Field Service representative.

BAD - ddnn: Handler/Driver missing

Explanation: The disk driver is not loaded.

User Action: Load the disk driver and reenter the command line.

ANALYZE/MEDIA (Cont.)

BAD - ddnn: Home block not found

Explanation: The ANALYZE/MEDIA command was unable to read the home block while attempting to validate the volume label. This message applies only to the /BADBLOCKS qualifier.

User Action: The disk must be initialized using the INITIALIZE command.

BAD - Invalid block number - numb

Explanation: You entered an invalid block number sequence. The value numb is the invalid sequence.

User Action: Type another value and reenter the command line. This message applies to the /BADBLOCKS qualifier only.

BAD - Invalid switch

Explanation: The message is self-explanatory.

User Action: Reenter the command line with a proper switch.

BAD - ddnn: Not in system

Explanation: The requested device does not exist on the RSX-11M configuration.

User Action: Ensure that you entered the command line correctly and specified the right device.

BAD - ddnn: Not ready

Explanation: The unit had not reached operating speed when BAD attempted to access it.

User Action: Allow the unit to reach operating speed and reenter the command line.

BAD - ddnn: Privilege violation

Explanation: An operation was attempted for a device that was mounted or allocated to another user.

User Action: Allocate another device, mount the device foreign, and reenter the command line.

BAD - Syntax error

Explanation: The ANALYZE/MEDIA command detected a syntax error on the command line.

User Action: Determine the correct syntax and reenter the command line.

ANALYZE/MEDIA (Cont.)

BAD - ddnn: Total bad blocks = n.

Explanation: This is an information message indicating the total number (decimal) of bad blocks on the volume.

User Action: Write the number of bad blocks on the label of the physical volume.

BAD - ddnn: Unrecoverable error n.

Explanation: An I/O error caused BAD to terminate. The value n is the error code number of the I/O error returned by the driver.

User Action: See the RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual for an explanation of the error code number.

BAD - ddnn: Vector not multiple of four

Explanation: The message is self-explanatory.

User Action: Reenter the command line including the /VEC switch with the proper value.

BAD - ddnn: Volume label incorrect

Explanation: The volume label entered with the /ALLOCATE switch did not match the label on the disk.

User Action: Reenter the command line using the correct volume label.

BAD - ddnn: Write locked

Explanation: The unit is write locked.

User Action: Write-enable the unit and reenter the command line.

ASSIGN**5.3 ASSIGN**

ASSIGN associates a logical name with a physical device, pseudo device, or other logical device. All references to the logical name are resolved by the system to the associated physical device, pseudo device, or logical device.

Format

```
ASSIGN[/qualifier[s]]
Device? ddnn:
Logical device? logicalddnn:
```

```
ASSIGN[/qualifier[s]] ddnn: logicalddnn:
```

Command Qualifiers

```
/GLOBAL
/LOCAL
/LOGIN
/SYSTEM
```

Parameters**ddnn:**

Specifies the name of the physical device, pseudo device, or logical device that is to receive the logical name.

logicalddnn:

Specifies the logical device name to be given to the physical device, pseudo device, or logical device.

The logical device name consists of a 2-letter mnemonic followed by an octal number terminated with a colon (:). There is no restriction on logical device names. They may use standard device mnemonics, such as DK0: or SP0:, or random letters, such as AZ1:.

Command Qualifiers

```
/GLOBAL
/SYSTEM
```

Specifies that the assignment is to be a global assignment. /SYSTEM and /GLOBAL are synonyms and are privileged qualifiers. Global assignments apply to all tasks running in the system.

/LOCAL

Specifies that the assignment is to be a local assignment. This is the default qualifier, so you do not need to specify it.

Commands and tasks initiated from your terminal can access devices through the logical names assigned to them. Note that no automatic deassignment occurs if you dismount a device after assigning a logical name to it.

You can define your set of logical assignments for your terminal. These logical names exist only for your terminal.

ASSIGN (Cont.)

/LOGIN

Specifies that the assignment is to be a login assignment. This is a privileged qualifier.

Login assignments are usually established through ACNT, the Account File Maintenance Program. However, the ASSIGN/LOGIN command does not alter the account file. These logical names exist for your account and are available to you regardless of the terminal on which you log in.

When a user issues a LOGIN command to log in on the system, the system automatically assigns the logical name SY0: to the user's system device (assuming it is different from the system device SY:), which is the device that contains the user's files.

Examples

```
$ ASSIGN (RET)
Device? DB1: (RET)
Logical device? TP1: (RET)
```

This example assigns the logical name TP1: to the physical device DB1:. The user may now issue commands referring to device TP1: in any command that accepts a device specification and DB1: will be substituted for it.

```
$ ASSIGN LB0: RR2: (RET)
```

This example assigns the logical name RR2: to the pseudo device LB0:.

```
$ ASSIGN/GLOBAL DK1: XX1: (RET)
```

This example assigns the logical name XX1: to the physical device DK1:. All users and tasks on the system can refer to XX1: when they initiate commands and tasks. This command must be issued from a privileged terminal. DK1: can also be used to reference DK1:.

Notes

The order of precedence in logical names, from highest to lowest, is: task, local, login, group, and global. This means that if the logical device SY0: has a global assignment of DB1: but a local assignment of DR0:, the operating system interprets SY0: to be DR0: for your terminal.

ASSIGN is counteracted by DEASSIGN. See Section 5.4.

You can display current assignments with SHOW ASSIGNMENTS, described in Section 5.11.1.

You can assign SY0: with SET DEFAULT, described in Section 4.6.1.

ASSIGN (Cont.)

ASSIGN/REDIRECT is a separate command, described in Section 5.3.1.

ASSIGN/TASK is a separate command, described in Section 5.3.2.

ASSIGN/QUEUE is a separate command, described in the RSX-11M/M-PLUS Batch and Queue Operations Manual.

ASSIGN/REDIRECT

5.3.1 ASSIGN/REDIRECT

ASSIGN/REDIRECT redirects output from one physical device to another. You can also redirect a physical device to a pseudo device, or the reverse. Use this privileged command when a device is unavailable and you have another device that can take its place. The devices need not be of the same type.

Format

```
ASSIGN/REDIRECT
From? oldddnn:
To? newddnn:
```

```
ASSIGN/REDIRECT oldddnn: newddnn:
```

Parameters

olddnn:

Specifies the device from which requests have been redirected.

newddnn:

Specifies the device to which subsequent requests for oldddnn: are to be redirected.

Example

```
$ ASSIGN/REDIRECT DT0: DT3: (RET)
$ SHOW DEVICES TYPE DT: (RET)
DT0:      DT3: LOADED
DT1:      LOADED
DT2:      LOADED
DT3:      LOADED
.
.
.
$ ASSIGN/REDIRECT (RET)
FROM? DT0: (RET)
TO? DT0: (RET)
$ SHOW DEV DT: (RET)
DT0:      LOADED
DT1:      LOADED
DT2:      LOADED
DT3:      LOADED
```

This example shows the process of redirection. Following the first ASSIGN/REDIRECT, any output for DT0: is sent to DT3:, as indicated by the first SHOW DEVICES display. Later, when DT0: is again available, the user issues ASSIGN/REDIRECT again to redirect DT0: to DT0:.

ASSIGN/REDIRECT (Cont.)**Notes**

Redirection does not affect any I/O requests already in the I/O queue.

The following restrictions apply to this command:

- You cannot redirect from or to a logical device name.
- You cannot redirect from one pseudo device to another pseudo device. You can, however, redirect a pseudo device to a physical device or the reverse.
- You cannot redirect pseudo device TI:.
- You cannot redirect an attached or spooled device.
- You cannot redirect a device unit containing a mounted volume.
- You cannot redirect a logged-in terminal to a device other than another terminal.

Error Messages

A** -- Circular redirect error

Explanation: Command attempted to establish a circular list of redirections.

User Action: Check available devices with SHOW DEVICES and set up legal redirection list. Reissue the command.

A** -- Device not redirectable

Explanation: Command attempted to redirect a device that cannot be redirected.

User Action: Check command for proper device names and reenter.

A** -- New device not known to system

Explanation: An ASSIGN/REDIRECT attempted to redirect to a new device that is not in the current system.

User Action: Check the devices on the system with SHOW DEVICES. Retype the command after checking for proper device names.

A** -- Old device attached

Explanation: The old ddnn: specified in the command is attached or spooled.

User Action: Find the cause of the error and proceed accordingly.

ASSIGN/REDIRECT (Cont.)

A** -- Old device mounted

Explanation: The oldddnn: specified in the command contains a mounted volume.

User Action: Wait for device to become available.

A** -- Old device not known to system

Explanation: An ASSIGN/REDIRECT attempted to redirect an unknown device.

User Action: Check devices with SHOW DEVICES. Reenter command after checking for proper device names.

A** -- Pseudo device redirect error

Explanation: Command attempted to redirect a pseudo device to another pseudo device.

User Action: None. You cannot redirect a pseudo device to another pseudo device.

A** -- TI redirect error

Explanation: Command attempted to reedirect pseudo device TI:.

User Action: None. TI: cannot be redirected.

ASSIGN/TASK**5.3.2 ASSIGN/TASK**

ASSIGN/TASK reassigns an installed task's logical unit numbers (LUNs) from one physical device to another. The reassignment overrides the static LUN assignments in the task's disk image file. This is a privileged command.

Format

```
ASSIGN/TASK
Task? taskname
Device? ddnn:
Logical Unit? lun
```

```
ASSIGN/TASK:taskname ddnn: lun
```

Parameters

taskname

Identifies the installed task whose LUN you wish to reassign.

ddnn:

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.

Examples

```
$ ASSIGN/TASK:NIKNAK (RET)
Device? DB1: (RET)
Logical unit? 5 (RET)
```

This example overrides the assignment of LUN 5 established at link time and reassigns LUN 5 to DB1:.

```
$ ASSIGN/TASK:PADWAK TT1: 4 (RET)
```

This example overrides the assignment of LUN 4 established at link time and reassigns LUN 4 to TT1:.

```
$ ASSIGN/TASK:GIVBON AZ1: 3 (RET)
```

This example overrides the assignment of LUN 3 established at link time and reassigns LUN 3 to logical device name AZ1:.

Notes

ASSIGN/TASK has no effect on a memory-resident task. Thus, this command has no effect on currently executing tasks or tasks fixed in memory. You can use ASSIGN/REDIRECT for dynamic device reassignments.

ASSIGN/TASK (Cont.)

Error Message

A** -- Lun out of range

Explanation: Command attempted to reassign a LUN that was higher than the maximum number of LUNs allocated during the task link.

User Action: Issue SHOW TASK:taskname/LOGICAL_UNITS to determine how many LUNs were allocated.

DEASSIGN**5.4 DEASSIGN**

DEASSIGN deletes logical-device assignments. DEASSIGN disassociates logical names from physical device names, pseudo device names, or logical device names assigned by ASSIGN.

Format

```
DEASSIGN[/qualifier[s]]
Logical device? logicalddnn:

DEASSIGN[/qualifier[s]] logicalddnn:
```

Command Qualifiers

```
/ALL
/GLOBAL
/LOCAL
/LOGIN
/SYSTEM
```

Parameter

logicalddnn:

Specifies the logical device name, assigned by ASSIGN, that is to be deassigned. This is a required parameter except with the /ALL qualifier. Logical device names have the same format as all other device names, a 2-letter mnemonic followed by an octal number terminated with a colon (:).

Command Qualifiers

/ALL

Deletes all assignments of the specified type. If you use /ALL with another qualifier, all assignments of the typed specified by the other qualifier are deleted. If this qualifier is used alone, all local assignments are deleted.

/LOCAL

Deletes a local logical name. This is the default.

/LOGIN

Deletes a login assignment. This is a privileged qualifier.

```
/GLOBAL
/SYSTEM
```

Deletes a global assignment. These two qualifiers are synonyms and are privileged.

Examples

```
$ DEASSIGN (RET)
Logical device? TP0: (RET)
```

This example deletes the assignment of logical device name TP0: to a device.

DEASSIGN (Cont.)

\$ DEASSIGN/ALL (RET)

This example deletes all local logical-name assignments.

\$ DEAS/LOCAL/ALL (RET)

This example is the equivalent of the previous example. /LOCAL is the default qualifier.

Notes

DEASSIGN counteracts ASSIGN.

You must type at least the first four characters of DEASSIGN.

Login assignments are normally established through ACNT, the Account File Maintenance Program.

You can display assignments with SHOW ASSIGNMENTS. See Section 5.11.1.

All local assignments disappear when you log out.

DISMOUNT does not delete logical name assignments.

DEASSIGN/QUEUE is a separate command. See the RSX-11M/M-PLUS Batch and Queue Operations Manual for more details.

ALLOCATE**5.5 ALLOCATE**

ALLOCATE declares a specified device to be a private device.

Format

```
ALLOCATE[qualifier[s]]
Device? dd[nn:] [logicalname]

ALLOCATE[/qualifier[s]] dd[nn:] [logicalname]
```

Command Qualifier

```
/TERMINAL:ttnn:
/TYPE:devicetype
```

Parameters

dd[nn:]

Specifies the device or device type to be allocated.

The device name can be entered in one of two forms:

1. If you use the form ddnn:, the specified device is allocated, if available.
2. If you use the form dd without the number or the colon, the first available device of the specified type is allocated. This is called generic allocation.

logicalname

Assigns a logical name to the device as you allocate it. This parameter is optional.

The logical name must be of the form xxnn:. Logical names take the same form as device names. The assignment is done automatically.

Command Qualifiers

```
/TERMINAL:ttnn:
```

Allocates the device to the specified terminal. This is a privileged qualifier.

```
/TYPE:devicetype
```

Specifies the type of device you wish to allocate. Use this qualifier with the generic allocation feature. If you issue an ALLOCATE command with the device name specified in the form dd without unit number or colon, the first available device with the name dd is allocated. Because devices with names beginning with DB can be RP04s, RP05s, or RP06s, and because the disk packs for these devices are not compatible, you could do a generic allocation specifying /TYPE:RP04 to allocate the first available RP04. See the examples.

ALLOCATE (Cont.)**Examples**

```
$ ALLOCATE DK (RET)
ALL -- DK0: Now allocated
```

This example allocates the first available RK05 drive. The message gives the name of the device allocated.

This is an example of a generic allocation.

```
$ ALLOCATE (RET)
Device? DK1: (RET)
$
```

This example allocates the specified device. The allocation is confirmed by the return of the prompt. No message.

```
$ ALL DK1: (RET)
ALL -- Device allocated to other user
```

This example shows an allocation that failed because the device named was not available.

```
$ ALL DB3: TP0: (RET)
ALL -- DB3: Now allocated
```

This example allocates the specified device and assigns it the logical name TP0:.

```
$ ALLOCATE/TYPE:RP04 DB (RET)
ALL -- DB3: Now allocated
```

This example shows the use of the /TYPE qualifier. The user wished to allocate the first available RP04 but wanted to ignore any other devices using the DB driver. This is also a generic allocation.

Notes

When you allocate a device, no other user can gain access to the device.

DEALLOCATE counteracts ALLOCATE. See Section 5.6.

If more than one user needs to access a volume on a device, the device should not be allocated. Each user should mount the volume. It remains mounted until all users who have mounted it have dismounted it.

ALLOCATE (Cont.)

Error Messages

ALL -- Device allocated to other user

Explanation: Command attempted to allocate device already allocated by another user.

User Action: Wait to allocate the device until the other user has deallocated it. (The terminal from which the device was allocated can be displayed by SHOW DEVICES.)

ALL -- Device attached

Explanation: Command attempted to allocate device attached by running task.

User Action: Wait to allocate device until task is finished.

ALL -- Pseudo device error

Explanation: Command attempted to allocate a pseudo device. Pseudo devices cannot be allocated.

User Action: Retype command after checking proper device name.

ALL -- Public device

Explanation: Command attempted to allocate a public device.

User Action: Public devices cannot be allocated.

ALL -- User logged on terminal

Explanation: Command attempted to allocate a terminal logged in by another user.

User Action: Only terminals that are not logged in can be allocated.

DEALLOCATE

5.6 DEALLOCATE

DEALLOCATE counteracts ALLOCATE. It frees a private device for access by other users.

Format

```
DEALLOCATE[/qualifier]
Device? ddnn:
```

```
DEALLOCATE[/qualifier] ddnn:
```

Command Qualifiers

```
/ALL
/DEVICE
```

Parameter

ddnn:

Specifies the device to be deallocated.

Command Qualifiers

/ALL

The /ALL qualifier frees all devices allocated from the terminal being used. If you use this qualifier, you cannot specify a device name.

/DEVICE

This qualifier is a non-operational.

Example

```
$ DEALLOCATE (RET)
Device? DB0: (RET)
```

This example deallocates the device DB0: previously allocated by the user.

Notes

Nonprivileged users can deallocate only those devices that they have previously allocated. Privileged users can deallocate any private device.

DEAL is the short form of DEALLOCATE.

Error Messages

DEA -- Device allocated to another user

Explanation: Command attempted to deallocate a device allocated by another user.

User Action: Nonprivileged users can only deallocate devices they have allocated from the same terminal. SHOW DEVICES displays the terminal from which the device was allocated.

DEALLOCATE (Cont.)

DEA -- Device mounted

Explanation: Command attempted to deallocate a mounted device.

User Action: You cannot deallocate a mounted device. Dismount the device.

MOUNT

5.7 MOUNT

MOUNT declares a volume to be logically known to the system, and sets the volume on-line and ready for use. Some qualifiers can be used with any MOUNT command; some are limited to mounting disks and other random-addressable devices, and others are limited to mounting magnetic tapes.

Format

For Disks and Other Random-Addressable Devices

```
MOUNT[/qualifier[s]] ddnn: volumelabel
```

For Magnetic Tapes

```
MOUNT[/qualifier[s]] ddnn:[,ddnn:...] fileset-ID
```

Command Qualifiers for Both Disks and Tapes

```
/FILE PROTECTION:(code)
/FOREIGN
/OVERRIDE:IDENTIFICATION
/PARAMETERS:"user parameters"
/PROCESSOR:arg
                acpname
                UNIQUE
/PROTECTION:(code)
/PUBLIC
/[NO]SHAREABLE
/[NO]SHOW
/SYSTEM (synonym for /PUBLIC)
/[NO]WAIT
/[NO]WRITE
```

Command Qualifiers for Disks and Other Files-11 Devices

```
/ACCESSED:n
/EXTENSION:n
/OWNER:[uic]
/UNLOCK
/WINDOW:arg
                n
                (USER:n,INDEX:n)
                FULL
```

Command Qualifiers for ANSI and Unlabeled Tapes

```
/BLOCK_SIZE:n
/CARRIAGE_CONTROL:arg
                FORTRAN
                LIST
                NONE
/DENSITY:arg
                800
                1600
```

MOUNT (Cont.)

```

/[NO]HDR3
/[NO]LABEL
/OVERRIDE: (arg[,s])
            ACCESSIBILITY
            EXPIRATION_DATE
            SET_IDENTIFICATION
/RECORD_SIZE:n
/TRANSLATE:arg
            EBCDIC
            NONE
            UT1
            UT2
            UT3
/VOLUME_IDENTIFICATION: (volume-ID[,volume-ID[,s]])

```

Parameters

ddnn:

Specifies device on which the volume is to be mounted. You can mount only one disk or other random-addressable device, but you can mount several magnetic tapes. See the discussion of mounting tapes and tape sets in Section 5.7.1.

NOTE

Disks and magnetic tapes differ in the way labeling is handled. See the separate discussions of volumelabel and fileset-ID. See also Section 5.7.1 for more on magnetic tape Volume IDs and File Set IDs.

volumelabel

Specifies the volume label, that is, the name associated with the volume. Volume labels are mandatory for nonprivileged users. You must supply a volume label for each volume you wish to mount. Disk and random-addressable volume labels can be as many as 12 characters.

Volume labels (or identifiers) can include any alphanumeric character without restriction. If you are mounting an ANSI-standard magnetic tape and if your system includes ANSI magnetic tape support, you can also include the following nonalphanumeric symbols:

```
<space> ! " % _ ' ( ) * + , - . / : < = > ?
```

Labels including these characters must be enclosed in quotation marks ("). If the label includes the quotation mark itself, it must be followed by another quotation mark. The extra quotation marks do not count in figuring the length of the label. See the examples.

Note that the at sign (@), semicolon (;), and dollar sign (\$) are not accepted in volume labels.

MOUNT (Cont.)

fileset-ID

Magnetic tape Volume Identifiers or File Set Identifiers can be as many as six characters in length. By default, the Volume Identifier on the first tape named in a MOUNT command is taken as the File Set Identifier as well. See the discussion in Section 5.7.1 for the distinction between magnetic tape File Set Identifiers and magnetic tape Volume Identifiers.

Command Qualifiers

/FILE_PROTECTION:(code)

Specifies the default file protection for any new files created on the volume while it is mounted. The file-protection code is enclosed in parentheses. See Section 4.1.6 for more information on file-protection codes and the relationship between file protection and volume protection. See also the /PROTECTION qualifier to MOUNT.

The default file protection can be overridden by specifying another protection when the file is created.

If this qualifier is not included, the value specified at the time the volume was initialized is applied.

For ANSI magnetic tapes, this qualifier specifies the protection applied to the volume. See the RSX-11M/M-PLUS and Micro/RSX I/O Operations Manual for further information on the translation from this coding to ANSI label coding.

/FOREIGN

Specifies that the volume being mounted is not in Files-11 format. Such volumes are called foreign volumes. Note that before a disk or tape has been initialized, it is a foreign volume.

The /FOREIGN qualifier is recommended but not required for mounting foreign volumes. If you do not mount the volume, access to the volume is unrestricted. If you mount a foreign volume /FOREIGN, no one else can access it. If you include /FOREIGN when mounting the foreign volume /SHAREABLE, then other users must also mount the volume to access it.

/OVERRIDE: IDENTIFICATION

Allows privileged users to mount a volume without using the volume label (or the File Set ID for magnetic tape). Identification is the default argument for /OVERRIDE.

If you are mounting a magnetic tape, you may want to combine the IDENTIFICATION argument with other /OVERRIDE arguments that are limited to use with magnetic tape. See the description of /OVERRIDE in the magnetic tape section.

MOUNT (Cont.)

`/PARAMETERS:"user parameters"`

Enables you to enter commands to an ACP that is not DIGITAL standard. MOUNT/PARAMETERS passes a maximum of 40 characters enclosed in quotation marks to a foreign ACP. You must use the /FOREIGN and /PROCESSOR qualifiers with this qualifier. If the quoted command includes the quotation mark itself ("), the quotation mark must be followed by another quotation mark. The extra quotation marks do not count in figuring the length of the command.

`/PROCESSOR:arg`
 acpname
 UNIQUE

Allows privileged users to specify the name of the ACP to use to support file activity on the volume. In most cases, you can use the default ACP. The default ACP depends on the device type.

If the device is a disk, MOUNT searches (in the order listed below) until it finds an ACP with a name in one of these forms:

- ddnnF1
- ddAACP
- F11ACP

Thus, for an RK05 device, which uses the device name DK:, MOUNT looks first for an ACP named DKnnF1, where nn is the device number. If no ACP of that name exists, MOUNT looks for an ACP named DKAACP and, finally, for F11ACP.

If the device is a DECTape, MOUNT searches (in the order listed below) until it finds an ACP with a name in one of these forms:

- ddnnF1
- ddAACP
- F11ACP

Thus, for a TU58 DECTape II, which uses the device name DD:, MOUNT looks for an ACP named DDnnF1. If no ACP of the appropriate name exists, MOUNT looks for an ACP named ddAACP. Failing that, MOUNT seeks an ACP named DDAACP and, finally, for the F11ACP.

If the device is a magnetic tape, MOUNT first looks for an ACP with a name in the form ddnnF1. Thus, for a TS04 device, which uses the device name MS:, MOUNT looks first for an ACP named MSnnF1, where nn is the device number. If no ACP of that name exists, MOUNT looks for an ACP named MSAACP and, finally, for the MTAACP.

MOUNT (Cont.)

You only have to specify an ACP in special cases. If an ACP has been written for a custom application in your installation, you will need to use that ACP for volumes used by that application. You will also have to include the /FOREIGN qualifier in this case. You may also need to use the /PARAMETERS qualifier to enter parameters used by that ACP.

The UNIQUE argument specifies that a dedicated ACP is to be used for the volume being mounted. MOUNT creates a copy of the FllACP if the device is a disk or DEctape, or if the MTAACP device is a magnetic tape. This unique ACP will be given a name of the form ddnnF1 or ddnnACP. When the volume is dismounted and the file processor exits, the ACP is removed automatically.

You can control the resources dedicated to file processing by choosing appropriate names for installing ACPs. For example, if you wish DB0: to have a dedicated ACP and for DB1: and DB2: to use a shared ACP, install the FllACP as DB00F1 and as DBAACP.

If you mount a volume /FOREIGN but do not specify an ACP, no ACP can access the volume. However, tasks such as INITIALIZE, DSC, BAD, or user tasks with non-file-structured device QIOS can access the foreign volume.

/PROTECTION:(code)

Specifies the volume protection for Files-11 disks. This protection overrides the volume protection established when the volume was initialized. The /PROTECTION qualifier combines with the /OWNER qualifier to control access to the volume. See Section 4.1.6 for more information on volume protection.

**/PUBLIC
/SYSTEM**

These two qualifiers are synonyms. The /SYSTEM qualifier is included for VAX/VMS compatibility.

Specifies that the mounted volume be available to all users who are allowed access under the volume-protection and file-protection codes established for the volume when mounted. See the discussion of file protection and volume protection in Chapter 4. These qualifiers are privileged.

If you state this qualifier explicitly when mounting a volume on an allocated (private) device, the device is automatically deallocated and set public.

If you do not state this qualifier explicitly, and the device is already set public, the mount will default to /PUBLIC.

See the discussion of public, private, and shareable volumes in Section 5.1.3.

MOUNT (Cont.)**/[NO]SHAREABLE**

Specifies whether the volume is to be mounted shareable.

A volume mounted /SHAREABLE can be mounted multiple times by the same or different users. See the discussion of multiple mounts in Section 5.1.3. Each user's access is determined by the volume-protection and file-protection codes established for the volume when mounted.

If you mount the volume /SHAREABLE and the device is allocated or set public, the device is automatically deallocated or set nonpublic.

A volume mounted /NOSHAREABLE is dedicated for your private use. No other user can access the volume. For Files-11 volumes mounted /NOSHAREABLE, your privileges are SYSTEM privileges. See the discussion of file protection and volume protection in Section 4.1.6.

If you specify /NOSHAREABLE explicitly when mounting a volume on a device, the device is allocated and set nonpublic, if necessary.

If the device is already allocated, the mount defaults to /NOSHAREABLE.

If the device is not allocated or set public, the default is /SHAREABLE.

If the device is set public, the default is /PUBLIC (synonym: /SYSTEM).

/[NO]SHOW

Specifies that information about the volume being mounted is to be displayed on your terminal.

The display format for disks and other random-addressable devices is as follows:

```

Volume Information
Class:          Files-11 or Foreign
Device:         ddnn:
Volume label:  label
Pack Serial:   serial number supplied by manufacturer
Owner:         [g,m]
Protection:    code
Default:       code
Processor:     acpname

```

The owner field indicates the owner of the volume. The UIC must be matched for protection checks for owner privileges. The owner UIC can be changed with the /OWNER qualifier. Protection indicates volume protection; Default indicates file protection.

The information displayed by /SHOW is similar for magtapes. If multiple tape volumes are being mounted, the device and volume label information is repeated for each volume. The Protection and Default fields are omitted from the volume information for magnetic tapes.

MOUNT (Cont.)**/[NO]WAIT**

Specifies whether you require operator assistance in performing the mount. The main use of this qualifier is for indirect command files, or terminals distant from the machine room. The default is /WAIT for mounts in indirect command files, and /NOWAIT for interactive mounts.

If the mount is included in an indirect command file, or if you specify /WAIT in an interactive mount, a message concerning the mount is sent to the operator's console and the mount is not completed until the operator physically places the disk on the drive and spins it up. If you specify /NOWAIT in a batch job or indirect command file, or if the mount is interactive, no message appears. The medium must have been placed on the device and readied for access (that is, spun up and on line). (See the examples.)

/[NO]WRITE

Specifies whether the volume is to be write-protected. The default is /WRITE. If /WRITE is specified or implied, the volume can be written to as permitted by the volume-protection and file-protection codes established for the volume when mounted. See the discussion of volume protection and file protection in Chapter 4. If /NOWRITE is specified, no one may write to the volume.

Command Qualifiers for Disks and Files-11 Devices**/ACCESSED:n**

Specifies the approximate number of directories that will be accessed simultaneously during the mount. Legal values for n range from 1 through 127. F11ACP maintains a list of most recently used directories. The value specified for /ACCESSED sets the number of entries in this list.

In general, you can significantly speed up directory operations by increasing this value. If a directory appears in the list, no disk I/O is required to find the directory in the MFD. In addition, the system avoids having to read the directory file header during directory operations. If this qualifier is not explicitly stated, the default is the value specified at the time the volume was initialized. Note that increasing the /ACCESSED value increases the amount of system pool used.

/EXTENSION:n

Specifies the number of blocks by which a file will be extended if it exhausts its space allocation. The value of n can be from 1 through 127. The default is the value specified when the volume was initialized.

/OWNER:[g,m]

Specifies the owner of the volume. The brackets are required syntax. This qualifier overrides the OWNER value established when the volume was initialized. The owner value is used with file and volume protection.

MOUNT (Cont.)**/UNLOCK**

Specifies that the volume index file, [0,0]INDEXF.SYS, is unlocked. This means the file can be read and written to. Normally, this file is locked and can be read only.

This qualifier has nothing to do with hardware or software write-locking.

/WINDOW:arg

n
(USER:n,INDEX:n)
FULL

Specifies the default number of mapping pointers to be allocated for file windows when the file is accessed. Legal values for n range from 1 through 127 or the word FULL. A file window consists of a number of mapping pointers stored in memory when the file is opened. If you specify FULL, windows are created that map as much of the file as possible.

You can specify different values for the user default and index file. If separate values are specified, the volume default will be the same as the user default specified and the index file window is created with the value specified.

The default for /WINDOWS is the value specified in the Volume Home Block. This value is established when the volume is initialized. If you wish to change that value, see the INITIALIZE/UPDATE command for information on altering the Volume Home Block.

Command Qualifiers for ANSI and Unlabeled Tapes**/BLOCK_SIZE:n**

Specifies the block size in characters for unlabeled magnetic tapes at the time of the mount. Values for n can be from 18 through 16000. This qualifier also works on ANSI Level 1 or 2 magnetic tapes, that is, those without HDR2 records (labels).

/CARRIAGE_CONTROL:arg

FORTTRAN
LIST
NONE

Specifies the record type for an unlabeled tape. The arguments correspond to the FORTRAN keywords for the OPEN statement. This qualifier also works on ANSI Level 1 or 2 magnetic tapes, that is, those without HDR2 records (labels). This qualifier also works for ANSI Level 3 magnetic tapes that do not have a system code (HDR1 CP 61) beginning in DEC or OS.

MOUNT (Cont.)

```
/DENSITY:arg
      800
     1600
```

Specifies the density at which data is to be written to the tape. The default is the lowest density supported by the specified tape unit. If the specified tape unit does not support the density given, the command is rejected.

```
/[NO]HDR3
```

Specifies whether an HDR3 label is to be written when creating new files. The /NOHDR3 qualifier is needed only in the situation where it is known that the target system cannot handle a tape with a HDR3 label. This qualifier has no other effect or application. The default is /HDR3.

```
/[NO]LABEL
```

Specifies whether the tape volume is ANSI or unlabeled. An ANSI tape is specified by the /LABEL qualifier. This is the default. An unlabeled tape is specified by the /NOLABEL qualifier.

If you mount a tape with the /NOLABEL qualifier specifying an odd record and block size, and then write a file to the tape using PIP, the block size will be the odd value specified by the MOUNT command plus 1. In other words, another byte is added to the block of data. This byte is the octal value 136 or circumflex (^). The circumflex character is added because some tape drives do not allow an odd number of bytes to be transferred to or from the tape. Therefore, the blocks of bytes are padded with the circumflex to allow data to be transferred to a tape on any tape drive.

For RMS-11 tape operations, the following occurs:

- The block size that you supply for writing is rounded up to the next long-word boundary (4-byte boundary). Any size block, including odd, will be read.
- If RMS-11 tries to write a short block (less than 18(decimal) bytes) because the data did not fill the block, RMS-11 writes an even-aligned (even-byte boundary) block of 18(decimal) bytes.
- In the case of blocks less than 18(decimal) bytes, RMS-11 fills the remainder of the block with circumflex characters.
- When short blocks are read, your user-written program is expected to recognize and discard the circumflex characters.
- RMS-11 ignores all circumflex records that have a FIXED format.

```
/OVERRIDE: (arg[,s])
      ACCESSIBILITY
      EXPIRATION DATE
      IDENTIFICATION
      SET_IDENTIFICATION
```

MOUNT (Cont.)

Specifies that you wish to override one or more characteristics of a tape volume that are normally checked when you mount a magnetic tape. The parentheses are optional if you wish to specify only one argument. If you wish to specify more than one argument, the arguments must be enclosed in parentheses and separated by commas.

ACCESSIBILITY - Specifies that you wish to override the protection on ANSI magnetic tapes.

The **EXPIRATION_DATE** - Specifies that you wish to override the expiration date on ANSI magnetic tapes.

The **IDENTIFICATION** - Specifies that you wish to mount the tape or tape set without specifying a File Set ID. By default, the File Set ID is the Volume ID of the first tape reel in the set. See Section 5.7.1.

The **SET_IDENTIFICATION** - Specifies that you wish to mount a tape or tape set that does not follow ANSI standard. This argument is not otherwise needed.

This is a privileged qualifier.

/RECORD_SIZE:n

Specifies the record size in characters for unlabeled magnetic tapes. The value for n can be from 1 through the value for blocksize on the volume.

/TRANSLATE:arg

EBCDIC
NONE
UT1
UT2
UT3

Specifies the character translation to take place for reading from and writing to the mounted unlabeled tape volume. NONE specifies no translation and is the default. EBCDIC specifies translation from ASCII to EBCDIC for writes and from EBCDIC to ASCII for reads. Note that the translation is not one-to-one because some characters are found only in EBCDIC or only in ASCII. UT1, UT2, or UT3 specify user-supplied translation tables that have been previously linked into the MTAACP. See the RSX-11M/M-PLUS and Micro/RSX I/O Operations Manual for further information on character translation.

/VOLUME_IDENTIFICATION:(volumeID[,volumeID[,s]])

Specifies Volume Identifiers. If you wish to check Volume IDs, you must specify this qualifier. If the File Set ID is not the same as the Volume ID of the first tape in the set, you must use this qualifier and include all Volume IDs in the set. See Section 5.7.1 for more information and for examples of this qualifier in use.

MOUNT (Cont.)

Examples

```
$ MOUNT (RET)
Device? DR1: (RET)
Volume ID? HOTROD (RET)
```

This example mounts the volume labeled HOTROD on device DR1:. If you are doing nothing more than mounting a volume to read from it or write to it, you will probably not need a more complicated MOUNT command than this.

All default values for the mount come from the Volume Home Block. This block is written when the volume is created by INITIALIZE or by some other system task, such as a backup utility, that creates a file structure on the volume. Most of the qualifiers to MOUNT are used to override values in the Volume Home Block.

```
$ MOUNT/DENSITY:1600 (MM0:,MM1:,MM2:) "W*MB*T",FOOBAR,FOO (RET)
```

Specifies that a 3-volume ANSI magnetic tape set be mounted on devices MM0:, MM1:, and MM2:. The volume labels are W*MB*T, FOOBAR, and FOO. The first label is included in quotes to allow specification of the asterisk (*). Note that the comma between "W*MB*T" and FOOBAR is outside the quotation marks.

```
$ MOUNT/PROCESSOR:MRXACP/FOREIGN/PARAMETERS:"MUMBLFRAZ" (RET)
Device? DR0: (RET)
Volume label? HARPO (RET)
```

This example mounts the volume labeled HARPO on device DR0:. The volume's file structure is controlled by a non-DIGITAL ACP named MRXACP, so the ACP must be named and the volume mounted foreign. This ACP accepts special commands. These are entered using the /PARAMETERS qualifier with the command string in quotation marks.

```
$ MOUNT/NOLABEL/BLOCK:2048/RECORD:64/CARR:FORTRAN/TRANSL:EBCDIC MM1: (RET)
```

The example mounts an unlabeled magnetic tape on device MM1:, specifying a block size of 2048 characters and a record size of 64 characters. The volume includes FORTRAN carriage control, so that is specified. The volume's records are written in EBCDIC characters, so the user included the /TRANSLATE qualifier. The characters will be translated to ASCII for handling by the operating system and translated back to EBCDIC for writing to the tape volume.

```
$ MOUNT/WAIT DB1: WHIZZER (RET)
```

The user wishes an operator to place the medium containing the proper volume in place on DB1:. The command may appear in an indirect command file, or interactive mount, perhaps from a user on a remote terminal. The following message appears on CO:, the operator's console:

```
CO:      Please mount volume WHIZZER on DB1:
Type
UNS (DCL START) when ready
or
RES (DCL CONTINUE) to reject request.
```

MOUNT (Cont.)

The operator performs the requested operation. The mount takes place if the device was available and the operator typed START (MCR UNS). If the device was not available and the operator typed CONTINUE (MCR RES), the mount fails. If the command appeared in an indirect command file, Indirect attempts to continue the operation.

Notes

You should not confuse mounting with physically placing the disk or other magnetic medium on a drive. Naturally, you must place the medium on a drive and spin it up (for a disk) or load it (for a tape) before you can do anything else with it, but mounting is enabling the system software to access the medium, either to read or write data, or to establish a file structure for the volume.

Mounting a device resets the error counts and QIOs to zero.

If you attempt to mount a volume on a device where the disk is not properly spun up, MOUNT will wait 30 seconds and try the mount again. If the disk is not spun up after 30 seconds, the mount fails. This feature has nothing to do with the /WAIT qualifier.

There are four states of device ownership:

Private	The device has been made private through the ALLOCATE command or the /NOSHAREABLE qualifier to MOUNT. Only you can access the volume.
Shareable	The device has been mounted /SHAREABLE. Any user can mount the volume.
Public	The device has been made public through the SET DEVICE/PUBLIC command or the /PUBLIC (or /SYSTEM) qualifier to MOUNT. Only a privileged user can set a device public.
Unowned	No one has mounted a volume on the device, nor is it allocated or set public.

See the discussion of public, private, unowned, and shareable devices and mounted volumes in Section 5.1.3.

You cannot put comments in a MOUNT command line.

DISMOUNT counteracts MOUNT. See Section 5.8 for a description of DISMOUNT with examples.

Some file attributes and positioning information for magnetic tape volumes or tape sets can be supplied after mounting using the MAG utility. For more information, see the RSX-11M/M-PLUS I/O Drivers Reference Manual.

See the introduction to this chapter for more information on how the operating system deals with devices and volumes. See also the description of the INITIALIZE command in Section 5.9.

MOUNT (Cont.)

5.7.1 Mounting Magnetic Tapes and Tape Sets

Mounting of magnetic tape volumes is handled by a special ACP, the MTAACP. This ACP handles not only magnetic tapes that meet the ANSI standard but also the most commonly encountered unlabeled tape formats.

The MOUNT command can only mount a single random-addressable volume, but magnetic tapes can be mounted by the set. In fact, when you mount a single magnetic tape, you are mounting a set consisting of only one tape. A magnetic tape set consists of logically related records written on one or more individual reels of tape. Only the first tape in the set need be loaded on a device at the time the MOUNT command is issued. However, all tapes in the set can be loaded on different devices at the time the MOUNT command is issued.

Because you mount magnetic tapes by the set, the terminology used for mounting magnetic tapes is somewhat different from that used in mounting disks. The tape set is identified by a File Set Identifier (File Set ID). Each individual tape in the file set is identified by a Volume Identifier (Volume ID). By default, the File Set ID is the same as the Volume ID of the first individual reel of tape in the set.

If you wish to use a File Set ID other than the Volume ID of the first reel in the set, you can do so on the first mount of a newly initialized tape as shown in the examples.

Each site using magnetic tapes extensively will have policies on File Set IDs and Volume IDs. See your system manager or supervisor for information on how File Set IDs and Volume IDs are used at your site.

The following examples establish the relationship between File Set IDs and Volume IDs.

The following command:

```
$ MOUNT MM0: FOO (RET)
```

mounts a tape set (consisting of a single physical reel) whose File Set ID is FOO and whose Volume ID is FOO. The reel FOO must be loaded on device MM0: at the time of the mount.

The following command:

```
$ MOUNT MM0: DRAG/VOLUME_IDENTIFICATION:(FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is DRAG) that consists of three physical reels with Volume IDs of FOO, BAR, and MORE. The reel with the Volume ID FOO must be loaded on device MM0:, but the other two reels are not loaded at the time of the mount. The File Set ID of the tape set was established with the first mount of the newly initialized tape set, using this same command. If this were the first mount, the File Set ID DRAG would be established by the MOUNT command, but the actual change of the File Set ID (from the default of FOO) does not take place until the first record is written to the tape.

MOUNT (Cont.)

The following command:

```
$ MOUNT/OVERRIDE:SET_IDENTIFICATION MM0: (FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is immaterial) that consists of three physical reels with Volume IDs of FOO, BAR, and MORE. The reel FOO must be loaded on device MM0:, but the other two reels are not loaded at the time of the mount.

The following command:

```
$ MOUNT MM0: FOO /VOLUME_ID:(A312,A387,B231) (RET)
```

mounts a tape set (whose File Set ID is FOO) that consists of three physical reels with Volume IDs A312, A387, and B231. The reel A312 must be loaded on device MM0:, but the other two reels are not loaded at the time of the mount.

The following command:

```
$ MOUNT (MM0:,MM1:,MM2:) (FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is FOO) that consists of three physical reels with Volume IDs FOO, BAR, and MORE. At the time of the mount, the reel FOO must be loaded on device MM0:, the reel BAR must be loaded on device MM1:, and the reel MORE must be loaded on device MM2:.

NOTE

There is no theoretical limit to the number of physical tape reels in a set, but your DCL MOUNT command must translate to an MCR MOU command of 80 characters or fewer. There are, therefore, mounts that are permitted by the system but that cannot be translated successfully from DCL to MCR within that limit. In those cases, you cannot use the DCL MOUNT command but must use the MCR MOU command. Remember, you can use the MCR command to enter MCR commands from a DCL terminal.

Error Messages

MOU -- ACP not in system

Explanation: The task specified as the ACP, or the default ACP, is not installed in the system.

User Action: See your system manager.

MOU -- Device not in system -ddnn:

Explanation: The command specified a device not present in the system.

User Action: Retype command after checking device list (SHOW DEVICES) or see your system manager.

MOUNT (Cont.)

MOU -- Device specified twice

Explanation: The command specified the same device twice.

User Action: Retype command after checking proper syntax.

MOU -- Driver not loaded

Explanation: The command named a device whose driver is not loaded.

User Action: See your system manager.

MOU -- Failed to attach device -ddnn:

Explanation: The command named a device that was attached by another task and could not be mounted.

User Action: Check device list (SHOW DEVICES) and find out if the device is in use. See your system manager.

MOU -- Home block I/O error

Explanation: An I/O error was detected in trying to read the Volume Home Block. This message often indicates that the device is not ready or that the disk has not been initialized.

User Action: Wait until the disk is spinning at full speed and try again.

MOU -- Illegal keyword combination

Explanation: The command specified conflicting qualifiers.

User Action: Retype command after checking proper syntax.

MOU -- Index file I/O error

Explanation: MOUNT could not read either the index file header or the storage allocation file.

User Action: See your system manager.

MOU -- No such device available

Explanation: The command named a device not present in the system.

User Action: Retype command after checking device list (SHOW DEVICES) or see your system manager.

MOU -- Not file-structured device

Explanation: The command named a device that is not supported as a Files-11 device, including ANSI magnetic tape.

User Action: Retype command after checking for proper device name.

MOUNT (Cont.)

MOU -- Parameter conflicts with mounted volume

Explanation: An attempt was made to mount a previously mounted volume using qualifiers that conflict with those specified when the volume was originally mounted.

User Action: Check with SHOW DEVICE and confirm qualifiers with other user. See your system manager.

MOU -- Storage bit map file I/O error

Explanation: An I/O error was encountered while reading the storage allocation file.

User Action: Check to be sure you have the correct magnetic medium, or the volume has not been initialized. See your system manager.

MOU -- Task not ACP

Explanation: The task named with the /ACP qualifier does not have the characteristics of an ACP.

User Action: Check to be sure you have named the right task.

MOU -- Undefined density selection

Explanation: The command specified an illegal density value.

User Action: Retype command after checking proper syntax.

MOU -- Unit allocated to or in use by another user

Explanation: The command specified a device that is already in use.

User Action: Check on the status of the device with SHOW DEVICES and take the appropriate action.

MOU -- Unit does not support density switch

Explanation: The command included a /DENSITY qualifier but named a device that does not accept it.

User action: Retype command after checking proper syntax.

MOU -- Unit does not support 800 BPI

Explanation: The command specified a device that does not support the 800 BPI density.

User Action: Retype command after checking proper syntax.

MOU -- Unit does not support 1600 BPI

Explanation: The command specified a device that does not support the 1600 BPI density.

User Action: Retype command after checking proper syntax.

MOUNT (Cont.)

MOU -- Unsupported file header format

Explanation: The Volume Home Block does not conform to Files-11 format. Volume may be corrupted.

User Action: Volume could be mounted with the /FOREIGN qualifier, or see your system manager.

MOU -- Volume already mounted -ddnn:

Explanation: An attempt was made to mount a volume on a device that already had a mounted volume. In multiuser protection systems, only random-access devices can be mounted by multiple users.

User Action: See your system manager.

MOU -- Wrong volume label

Explanation: The volume label on the label and the volume label in the command do not match.

User Action: Retype command after checking for proper volume label. Privileged users can use the /OVERRIDE:IDENTIFICATION qualifier.

DISMOUNT**5.8 DISMOUNT**

DISMOUNT marks the volume mounted on the specified device to be logically off line and disconnected from the file system. Marking a volume for dismount prevents programs from opening new files on the volume. After all open files on the volume have been closed, the ACP dismounts the volume.

Format

```
DISMOUNT[/qualifier[s]]
Device? ddnn: [volumelabel]
```

```
DISMOUNT[/qualifier[s]] ddnn: [volumelabel]
```

Command Qualifiers

```
/ALL
/PUBLIC
/SAVE
/SYSTEM (synonym for /PUBLIC)
/TERMINAL:ttnn:
/[NO]UNLOAD
```

Parameters**ddnn:**

Specifies the device on which the volume is mounted.

volumelabel

Specifies the volume label or File Set ID for magnetic tape. This parameter is optional, but if it is specified, the label or File Set ID is checked against the mounted volume. The volume label or File Set ID can be a quoted string, following the same rules as explained in the discussion of MOUNT, Section 5.7.1. There is no prompt for this parameter.

Command Qualifiers**/ALL**

Specifies that all volumes mounted from the terminal at which the command is issued be dismounted. A message informs you of each dismount as it takes place. No device name or volume label is accepted with this qualifier. See the example.

**/PUBLIC
/SYSTEM**

Causes all users to be dismounted from a volume. This is a privileged qualifier. A DISMOUNT/PUBLIC frees the device, no matter who has mounted the volume on it. This is the only way to dismount a public volume. /SYSTEM is a synonym for /PUBLIC, included for compatibility with VAX/VMS.

DISMOUNT (Cont.)**/SAVE**

Specifies that the magnetic tape is to be rewound to the first file position and to still be accessible for read or write operations by privileged tasks after the dismount.

This qualifier applies only to magnetic tape drives.

/TERMINAL:tttn:

Allows a privileged user to dismount a volume mounted from another terminal.

/[NO]UNLOAD

Specifies that the magnetic tape is to be completely rewound after the dismount. The /UNLOAD qualifier is the default.

The /NOUNLOAD qualifier rewinds the tape to the first file position. Remember, these values can be set with the /DEFAULT qualifier to MOUNT.

The /UNLOAD and /NOUNLOAD qualifiers apply to magnetic tape drives only.

Examples

```
$ DISMOUNT (RET)
Device? DK0: (RET)
```

```
DIS -- TT11: Dismounted from DK0: *** Final Dismount Initiated ***
```

This example dismounts a volume on device DK0:. No other user had the volume mounted when the DISMOUNT command was issued.

```
$ DISMOUNT DK1: HOTROD (RET)
```

```
DIS -- TT56: Dismounted from DK0:
```

This example dismounts a volume on device DK1:. The user specified the label HOTROD to be sure the correct volume was mounted on the device. The volume was mounted shareable and some other user also has it mounted, so the informational message does not include the notification of final dismount.

```
$ LOGOUT (RET)
DMO -- TT11: Dismounted from DX0: *** Final dismount ***
Have a good morning
22-MAR-85 TT11: Logged off
```

This example dismounts a volume as part of the LOGOUT procedure. The user had the volume mounted on the device and had not dismounted it before logging off. LOGOUT dismounted the volume. If the device was allocated, LOGOUT also deallocated it. This is the equivalent of DISMOUNT/ALL.

DISMOUNT (Cont.)

The informational message is headed by DMO rather than DIS because, at this point, DCL is no longer involved in the dismount.

```
$ DISMOUNT/ALL (RET)
DIS -- TT11: Dismounted from DK0:
DIS -- TT11: Dismounted from DK1: *** Final dismount ***
DIS -- TT11: Dismounted from MM2: *** Final dismount ***
```

This example dismounts all volumes mounted from the terminal at which the DISMOUNT was issued. See next example.

```
$ DISMOUNT/ALL (RET)
DIS -- TT11: Dismounted from DK0:
DIS -- TT11: Dismounted from DX1: *** Final dismount ***
DIS -- TT11: Dismounted from MM2: *** Final dismount ***
$ MOUNT DX1: CRUCIAL (RET)
```

This example dismounts all volumes mounted from the terminal at which the DISMOUNT was issued. The message reminds the user that a volume was mounted on DX1:. As this volume was necessary to a running task, the user immediately issued a new MOUNT command using the volume label CRUCIAL. So long as the task did not attempt to open a file and did not close its last file between the time of the message and the time of the new MOUNT, the DISMOUNT has no effect on the task. All files that the task had open remain open.

```
$ DISMOUNT/PUBLIC DB3: (RET)
DIS -- TT21: Dismounted from DB3:
DIS -- TT27: Dismounted from DB3:
DIS -- TT31: Dismounted from DB3:
DIS -- TT55: Dismounted from DB3: *** Final dismount ***
```

In this example, a privileged user dismounted all users who had mounted the volume on DB3:.

Notes

DISMOUNT counteracts MOUNT.

LOGOUT issues a DISMOUNT/ALL automatically. DISMOUNT/ALL dismounts all volumes mounted from the terminal, whether mounted explicitly through a MOUNT command or implicitly through a task that spawns a MOUNT command.

You cannot include comments in a DISMOUNT command line.

The messages from the ACP to your terminal indicate only that the volume is marked for dismount. The actual completion of the dismount is noted on the operator's console.

DISMOUNT (Cont.)

CAUTION

When the dismount operation is complete, the ACP prints the following message on the operator's console (CO:):

*** ddnn Dismount complete

This message does not appear until all files open on the volume are closed. Do not remove the medium from the drive until this message appears on CO:. If you remove the medium before the message appears, the present volume may be corrupted and the next volume mounted on that device will be corrupted.

Error Messages

DMO -- WARNING -- All units of multiunit drive will spin down <ddnn:>

Explanation: When a unit of a multiunit device is spun down (by a privileged user), all other units of the device are also spun down.

User Action: No action is required.

DMO -- Checkpoint file still active

Explanation: The command attempted to dismount a volume that contained an active checkpoint file. The volume cannot be dismounted until the checkpoint file has been closed.

User Action: Wait, or, if privileged, issue a SET DEVICE:ddnn:/NOCHECKPOINT and reissue the DISMOUNT after receiving the system message indicating that the checkpoint file is no longer active.

DMO -- Volume not mounted

Explanation: The command specified a device that was not mounted.

User Action: Retype command after checking SHOW DEVICES for mounted devices.

DMO -- Volume not mounted by TI:

Explanation: The command attempted to dismount a volume mounted from another terminal.

User Action: Dismount the volume from the proper terminal, or have a privileged user dismount the volume.

DMO -- Wrong volume label

Explanation: The command included an incorrect volume label.

User Action: Reissue the command without specifying a volume label or check the volume label to be sure you are dismounting the right volume.

INITIALIZE

5.9 INITIALIZE

INITIALIZE produces a volume in Files-11 format.

For disks and DECTapes, INITIALIZE destroys all existing files, writes a dummy bootstrap and a Volume Home Block, and builds the directory structure.

For magnetic tapes, INITIALIZE writes an ANSI standard volume label and a dummy file that destroys all existing files.

Many of the values specified in the INITIALIZE command can be changed with the INITIALIZE/UPDATE command without disturbing existing data on the volume. See Section 5.9.1. Volumes must be allocated and mounted foreign before they can be initialized.

Format

```
INITIALIZE[/qualifier[s]]
Device? ddnn:
Label? volumelabel

INITIALIZE[/qualifier[s]] ddnn: volumelabel (volume-ID)
```

Command Qualifiers

```
/ACCESSED:n
/BAD_BLOCKS:arg
    AUTOMATIC
    (AUTOMATIC,MANUAL)
    MANUAL
    NOAUTOMATIC
    OVERRIDE
    (OVERRIDE,MANUAL)
/DENSITY:arg
    800
    1600
    HIGH
    LOW
/EXTENSION:n
/FILE_PROTECTION:(code)
/HEADERS:n
/INDEX:arg
    BEGINNING
    MIDDLE
    END
    n
/LABEL:VOLUME_ACCESSIBILITY:"c"
/MAXIMUM_FILES:n
/OWNER:[g,m]
/PROFESSIONAL
/PROTECTION:(code)
/[NO]SHOW
/WINDOWS:n
```

Parameters

ddnn:

Specifies the name of the device on which the magnetic medium to contain the volume has been placed.

INITIALIZE (Cont.)

volumelabel

Specifies the label the volume is to be initialized with. The label names the volume and must be specified by nonprivileged users when they mount the volume. The volume label is in effect a password controlling access to the volume. For magnetic tapes, the situation is more complex. See Section 5.7.1.

For disks and DECTapes, volumelabel can be as many as 12 characters.

For magnetic tapes, Volume ID can be as many as six characters.

See Section 5.7, the MOUNT command description, for details on the characters that can be included in volume labels and Volume IDs.

Command Qualifiers

Selecting command qualifiers for INITIALIZE requires a good knowledge of the Files-11 file structure. You should see the RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual for further information.

Defaults to INITIALIZE enable you to initialize a volume in a standard fashion, but the qualifiers allow much more flexibility in using the system.

/ACCESSED:n

This qualifier specifies the number of directories that will be accessed simultaneously. F11ACP maintains a list of most recently used directories. If a directory is in this list, no disk I/O is required to find the directory in the Master File Directory, nor do directory operations require that the directory file header be read. In general, directory operations can be speeded up significantly by increasing the /ACCESSED count but at the potential cost of pool.

The default n is 3. Legal values for n range from 1 through 127.

/BAD_BLOCKS:arg

```
AUTOMATIC
(AUTOMATIC,MANUAL)
MANUAL
NOAUTOMATIC
OVERRIDE
(OVERRIDE,MANUAL)
```

AUTOMATIC reads the bad-block descriptor file created the ANALYZE/MEDIA command and determines the bad-block information for the volume automatically. This is the default.

(AUTOMATIC,MANUAL) reads the bad-block descriptor file and, when done, accepts specification of additional blocks entered from the terminal.

MANUAL accepts bad-block information for BADBLK.SYS entered from the terminal.

INITIALIZE (Cont.)

NOAUTOMATIC ignores the bad-block information and performs no bad-block processing.

OVERRIDE ignores the manufacturer's bad-block descriptor file written on the last track of the volume, reads the bad-block descriptor file created by the ANALYZE/MEDIA/OVERRIDE command on the last good block before the last track, and automatically determines the bad-block information for the volume. This argument is valid only for DL:, DM:, and DR: devices that contain manufacturer-recorded bad-block data in the last track.

(OVERRIDE,MANUAL) ignores the manufacturer's bad-block descriptor file written on the last track of the volume, reads the bad-block descriptor file created by the ANALYZE/MEDIA/OVERRIDE command on the last good block before the last track, automatically determines the bad-block information for the volume, and, when done, accepts specification of additional blocks entered from the terminal. This argument is valid only for DL:, DM:, and DR: devices that contain manufacturer-recorded bad-block data in the last track.

```
/DENSITY:arg
      800
      1600
      HIGH
      LOW
```

Specifies the density in bits per inch (bpi) for the medium loaded on the device. The argument can be 800, 1600, HIGH, or LOW, depending on the device type. Only volumes on media for certain devices can be initialized at different densities. You must mount the volume at the same density with which it was initialized.

The HIGH and LOW values can be checked on an existing diskette volume with the INITIALIZE/UPDATE command. See Section 5.9.1.

Here are the acceptable density values for DIGITAL devices with switchable density:

Mnemonic	Device	Densities
MM: (magnetic tape)	TE16	800 (default) 1600
MM: (magnetic tape)	TU16	800 (default) 1600
MM: (magnetic tape)	TU45	800 (default) 1600
MM: (magnetic tape)	TU77	800 (default) 1600
DY: (diskette)	RX02	LOW (or single) HIGH (or double) (default)

INITIALIZE (Cont.)**/EXTENSION:n**

Specifies the number of blocks by which the file is to be extended when its allotted space is exhausted. The default n is 5.

This value can be altered on an existing volume with the INITIALIZE/UPDATE command. See Section 5.9.1.

/HEADERS:n

Specifies the number of file headers to be allocated initially in the index file. The five system files INDEXF.SYS, BITMAP.SYS, CORIMG.SYS, BADBLK.SYS, and 000000.DIR are not included in the value for n.

The default value of n is proportional to the capacity of the medium; the minimum allocation is 16 headers.

The following table lists the default number of file headers allocated to each device.

Mnemonic	Device	File Headers Allocated Initially (n)
DB:	RP04/05	5283
DB:	RP06	10478
DD:	DEctape II	16
DK:	RK05	16
DL:	RL01	314
DL:	RL02	629
DM:	RK06	834
DM:	RK07	1654
DR:	RM02/03	4049
DR:	RM05	25593
DR:	RM80	7461
DS:	RS03/04	16
DT:	DEctape	16
DU:	RA60	12308
DU:	RA80	7314
DU:	RA81	51699
DU:	RC25/RCF25	3020
DU:	RD51	Varies
DU:	RD52	Varies
DU:	RD53	Varies
DU:	RX50	16
DX:	RX01	16
DY:	RX02	16
EM:	ML11	Varies

The maximum number of file headers that can be allocated to a volume is either determined by the value specified with the /MAXIMUM_FILES qualifiers or is defaulted. The default number is based on the size of the volume (see the description of the /MAXIMUM_FILES qualifier for the calculation). The maximum number of file headers determines if the volume will have a one-, two-, or three-header index file.

INITIALIZE (Cont.)

You need to determine the maximum number of the file headers allowed on the volume before you can calculate the number of headers that will be preallocated initially. The number of preallocated file headers is determined either by the value you specified for the /HEADERS qualifier or by a formula that uses the value you specified with /MAXIMUM_FILES. The following list gives three general formulas (one for each type of index file):

one-header index file

headers = maximum/2

headers The initial number of preallocated file headers

maximum The maximum number of files (specified with /MAXIMUM_FILES)

If headers is less than 100(decimal), INITIALIZE preallocates 16(decimal) file headers for your volume.

two-header index file

headers = maximum/2 or 25593(decimal), whichever is larger

three-header index file

headers = maximum/2 or 51699(decimal), whichever is larger

See the description of the /MAXIMUM_FILES qualifier for more information on multiheader index files.

/INDEX:arg

BEGINNING
MIDDLE
END
n

Specifies the location of the index file on the volume. This qualifier forces a specified location for the index file, the Master File Directory, and the storage allocation file. This is usually done to minimize access time.

Four possibilities are available for the argument field:

BEGINNING The beginning of the volume. This is the default for tapes and DECTapes.

MIDDLE The middle of the volume. This is the default for most disks.

END The end of the volume.

n Logical block n.

INITIALIZE (Cont.)

`/LABEL:VOLUME_ACCESSIBILITY:"c"`

Specifies the access protection for a magnetic tape volume.

INITIALIZE places the argument "c" in VOL1 (the accessibility field), which is usually left blank. Legal values for c are the uppercase letters A through Z, the digits 0 through 9, and the following special characters:

`<space> ! " % & ' (*) + , . / : < = > ? ;`

The default for c is a space.

Note that if any nonblank character is specified, only a privileged user will be able to mount the volume.

`/MAXIMUM_FILES:n`

Specifies the maximum number of files permitted on the volume. This number corresponds to the number of file headers in the volume's index file (see the description of the `/HEADERS` qualifier).

The value of n includes the five system files: INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, and 000000.DIR. The value of n must be less than or equal to the maximum number of files permitted on the volume. To determine the default maximum number of files for a disk size of X, use the following calculation:

$$\text{default-MAXIMUM_FILES} = ((X - (X + 4095.) / 4096.) + 9.) * 127. / 258.$$

The value also decides if the volume will be a Files-11 level 401 or 402 volume. Level 402 volumes will have a multiheader index file and allow up to 65,500 files on them. INITIALIZE creates the multiheader index file and modifies the volume's home block automatically if the value specified is greater than 25,593. (If 65K index file support is not specified for the INITIALIZE task, the largest value allowed for `/MAXIMUM_FILES` is 25,593. With 65K support, the largest value allowed is 65,500.) Note that such volumes are not backwards compatible from RSX-11M Version 4.1. INITIALIZE issues a warning message about incompatibility when it creates a structure level 402 volume.

The following table also includes the default number of files and the maximum number of files for each device.

Mnemonic	Device	Default n	Maximum n
DB:	RP04/05	10567	25600
DB:	RP06	20956	25600
DD:	DECTape II	30	247
DK:	RK05	294	2357
DL:	RL01	629	5034
DL:	RL02	1259	10074
DM:	RK06	1668	13344
DM:	RK07	3308	25600
DR:	RM02/03	8099	25600
DR:	RM02/03	8099	64798
DR:	RM05	30781	65500

INITIALIZE (Cont.)

Mnemonic	Device	Default n	Maximum n
DR:	RM80	14923	65500
DS:	RS03	62	499
DS:	RS04	125	1003
DT:	DEctape	34	278
DU:	RA60	24617	65500
DU:	RA80	14629	65500
DU:	RA81	54815	65500
DU:	RC25/RCF25	6040	48326
DU:	RD51	varies with size of disk	
DU:	RD52	varies with size of disk	
DU:	RD53	varies with size of disk	
DU:	RX50	48	388
DX:	RX01	29	238
DY:	RX02	60	481
EM:	ML11	varies with size of disk	

NOTE

The /MAXIMUM FILES value must be computed by the maximum number of user UFDs and files plus the five files (BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, INDEXF.SYS, and 000000.DIR) required for Files-11 structure. See the RSX-11M/M-PLUS and Micro/RX I/O Operations Reference Manual for more information on creating the disk structure.

This value can be altered on an existing volume with the INITIALIZE/UPDATE command. See Section 5.9.1.

/FILE_PROTECTION:(code)

Specifies the default protection for all files on the volume being initialized. See Section 4.1.6 for more information on file protection codes.

The default protection code is as follows:

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

This value can be altered on an existing volume with the INITIALIZE/UPDATE command. See Section 5.9.1.

/OWNER:[g,m]

Specifies the owner of the volume. Group and member numbers range from 1 through 377. The default owner is [1,1]. The /OWNER value is used for checking volume and file protection. See Section 4.1.6 for more information on volume and file protection.

/PROFESSIONAL

Applies different defaults to a volume when the volume is initialized. The qualifier assumes the volume is an RX50.

INITIALIZE (Cont.)

You cannot use this qualifier with any other qualifier except the /SHOW qualifier. When the /SHOW qualifier is also used, the system displays the following message:

```
Volume initialized with /PROFESSIONAL
```

All other information except the volume label is suppressed. The label is displayed only if you specified a label in the command line. If you do not specify a label, /PROFESSIONAL generates its own label for the volume: the first 12 characters of the date and time, excluding hyphens and colons. The second digit for the seconds is also dropped. For example:

```
20MAY8510453
```

When you use the /PROFESSIONAL qualifier, the system files (index, bitmap, bad block, core image, and directory) are located at the beginning of the volume instead of in the middle. The default directory for these files is [200,200] instead of [1,1]. These defaults cannot be changed.

Although no other qualifier except /SHOW can be used with /PROFESSIONAL, the qualifier also has the following effects: for /FILE_PROTECTION, the default access for all classes (including world) is RWED; for /MAXIMUM_FILES, the default number of files allowed on the volume is 200.

/PROTECTION:(code)

Specifies the default protection for new files created on the volume. See the discussion of volume protection in Section 4.1.6. The code for volume protection is similar to that for file protection, except that the "E" for EXTEND protection is replaced by a "C" for CREATE protection.

/[NO]SHOW

Specifies that you wish to see all the values with which the volume is being initialized. /NOSHOW is the default.

These values are identified in MCR terms. Here are the DCL equivalents:

Display	DCL Equivalent
Label	Label
/Bad	/BAD_BLOCKS
/Cha	
/Ext.	/EXTENSION
/FPro.	/FILE_PROTECTION
/Indx.	/INDEX
/Inf.	/HEADERS
/Lru.	/ACCESSED
/Mxf.	/MAXIMUM_FILES
/Pro.	/PROTECTION
/Uic.	/OWNER
/Win	/WINDOWS

INITIALIZE (Cont.)

/WINDOWS:n

Specifies the number of mapping pointers to be allocated for file windows. A file window consists of a number of mapping pointers stored in memory when the file is opened. The default for n is 7.

This value can be altered on an existing volume with the INITIALIZE/UPDATE command. See Section 5.9.1.

Examples

```
$ INITIALIZE (RET)
Device? DR0: (RET)
Volume ID? HONOLULU (RET)
```

This example initializes a volume with the volume label HONOLULU on device DR0:. See next example.

```
$ ALLOCATE DR0: (RET)
$ MOUNT/FOREIGN DR0: (RET)
$ INITIALIZE DR0: HONOLULU (RET)
$ DISMOUNT DR0: (RET)
$ MOUNT DR0: HONOLULU (RET)
```

This example illustrates the process necessary to turn a fresh disk into a volume in Files-11 format that you can use. (This example assumes bad block information from ANALYZE/MEDIA has already been written on the medium.) See Section 5.1.4 for a more detailed treatment of this process.

First the device is allocated, or made private, so that no other user can access the device during this process. Then MOUNT/FOREIGN provides access to the disk although it is not yet in Files-11 format. The volume is initialized and given the name HONOLULU. The volume is then dismounted. Finally, the volume is mounted conventionally using the volume label HONOLULU. The /FOREIGN qualifier is not required on RSX-11M.

```
$ INIT/INDEX:END/PROTECTION:(SYSTEM:RWED,OWNER: ,GROUP: ,- (RET)
->WORLD:) DUL: MIRACLE (RET)
```

This example initializes a volume named MIRACLE on a medium mounted on device DUL:. The index file is placed at the end of the volume. Files on the volume are accessible to the system and privileged users only. The user used the continuation character (-) to make room for the command on a narrow terminal.

Notes

Nonprivileged users must allocate the device on which the volume is to be initialized.

The medium need not be mounted to be initialized, but mounting with the /FOREIGN qualifier is recommended.

INITIALIZE (Cont.)

Error Messages

INI -- Allocation for sys file exceeds volume limit

Explanation: The system was unable to allocate a system file from the specified block because of intermediate bad blocks or end of volume.

User Action: Enter command with different argument for the /INDEX: qualifier.

INI -- Bad block file corrupt - data ignored

Explanation: Although the /BAD_BLOCKS qualifier was selected, or defaulted to, the bad block data on the disk was not in the correct format and was therefore ignored.

User Action: Process the medium with the BAD utility and initialize again.

INI -- Block(s) exceed volume limit

Explanation: The specified block or blocks exceeded the physical size of the volume.

User Action: Retype command after checking for proper qualifier values.

INI -- Boot block write error

Explanation: An error was detected in writing out the volume boot block.

User Action: Reenter the command. If it still does not work, see your system manager.

INI -- Checkpoint file header I/O error

Explanation: An error was detected in writing out the checkpoint file header.

User Action: Reenter the command. If it still does not work, see your system manager.

INI -- Command I/O error

Explanation: INITIALIZE encountered an I/O error while reading the command line.

User Action: Retype command line. If this fails, see your system manager.

INI -- Data error

Explanation: The command specified a bad block number or contiguous region that was too large.

User Action: Retype command after checking proper syntax.

INITIALIZE (Cont.)

INI -- Device allocated to other user - ddnn:

Explanation: Command specified a private device not allocated to your terminal.

User Action: Retype command after checking SHOW DEVICES for proper device name.

INI -- Device not in system

Explanation: Command specified a device not in the current system.

User Action: Retype command after checking SHOW DEVICES for proper device name.

INI -- Device not ready -ddnn:

Explanation: The device was not up to speed (spun up).

User Action: Wait and try again.

INI -- Driver not loaded

Explanation: Command specified a device for which the driver is not loaded.

User Action: Retype command after checking SHOW DEVICES for proper device name.

INI -- Failed to attach device -ddnn:

Explanation: INITIALIZE failed to attach the specified device.

User Action: Wait and try again. Check SHOW DEVICES. See your system manager.

INI -- Failed to read bad block file

Explanation: Although the /BAD_BLOCKS:AUTOMATIC qualifier was specified, or defaulted to, no bad-block data was found.

User Action: Process the medium through the BAD utility and try again.

INI -- Magtape label must be specified

Explanation: Command attempted to initialize a magnetic tape volume without specifying the required volume label.

User Action: Retype command after checking proper syntax. Magnetic tape volume labels are limited to six characters.

INI -- Magtape write error -ddnn:

Explanation: While writing to the magnetic tape, the system encountered an I/O error.

User Action: Retype command.

INITIALIZE (Cont.)

INI -- No bad block data found

Explanation: Although the /BAD_BLOCKS:AUTOMATIC qualifier was specified, or defaulted to, no bad-block data was found.

User Action: Process the medium through BAD utility and try again.

INI -- Not file structured

Explanation: The system does not support the device named as a Files-11 device.

User Action: Retype command after checking proper syntax.

INI -- Public device -ddnn:

Explanation: Command attempted to initialize a volume on a public device.

User Action: Nonprivileged users can only initialize volumes on private devices.

INI -- Storage bit map file error

Explanation: The system failed to read the header of the file [0,0]BITMAP.SYS.

User Action: Use the BAD utility on the disk again to find bad blocks, then try the operation again.

INI -- Undefined density selection

Explanation: Command specified an illegal density argument for the device named.

User Action: Retype command after checking for proper density argument.

INI -- Unit is not at high density

Explanation: Command specified that diskette in RX02 (DYnn:) drive be initialized at low density, but diskette was formatted at high density.

User Action: Reformat diskette, or change it.

INI -- Unit is not at low density

Explanation: Command specified that diskette in RX02 (DYnn:) drive be initialized at high density, but the diskette was formatted at low density.

User Action: Reformat diskette, or change it.

INITIALIZE (Cont.)

INI -- Volume mounted Files-11

Explanation: Command attempted to initialize a volume mounted Files-11.

User Action: Check to make sure you have the right volume.

INI -- Volume mounted foreign with ACP

Explanation: Command attempted to initialize a volume mounted /FOREIGN but with the /ACP qualifier.

User Action: Remount the volume without the /ACP qualifier.

INI -- Volume name too long -- volumelabel

Explanation: Command included a volume label that was too long.

User Action: Retype command, but limit volume label for disks or DECTapes to 12 characters; magnetic tapes to six characters.

INITIALIZE/UPDATE

5.9.1 INITIALIZE/UPDATE

INITIALIZE creates a Volume Home Block that includes a number of values used by the device's Ancillary Control Processor (ACP) when you mount the volume. INITIALIZE/UPDATE uses the HOME utility to alter some of those values without affecting the other data on the volume. The INITIALIZE/UPDATE command is only for disk and DECTape volumes in Files-11 format.

Format

```
INITIALIZE/UPDATE[/qualifier[s]] ddnn: volumelabel
```

Command Qualifiers

```
/ACCESSED:n
/DENSITY:arg
    HIGH
    LOW
/EXTENSION:n
/FILE PROTECTION:code
/LABEL:newvolumelabel
/MAXIMUM FILES:n
/OWNER:[g,m]
/PROTECTION:code
/[NO]SHOW
/WINDOWS:n
```

Command Qualifiers

/ACCESSED:n

Specifies the approximate number of directories that will be accessed simultaneously during the mount. Legal values for n range from 1 through 127. FllACP maintains a list of most recently used directories. The value specified for /ACCESSED sets the number of entries in this list.

In general, directory operations can be significantly speeded up by increasing this value. If a directory appears in the list, no disk I/O is required to find the directory in the MFD. In addition, directory operations take place without the necessity of reading the directory file header. Note that increasing the /ACCESSED value increases the amount of system pool used.

/DENSITY:arg
HIGH
LOW

Checks the /DENSITY value established when the volume was initialized. See the description of the /DENSITY qualifier under INITIALIZE (Section 5.9) for more information.

/EXTENSION:n

Specifies the number of blocks by which a file will be extended if it has exhausted its space allocation. The argument n can be from 1 through 127. The default is set by your system manager.

INITIALIZE/UPDATE (Cont.)

`/FILE_PROTECTION:code`

Specifies the protection for new files created on the volume. See Section 4.1.6 for more information on file protection.

`/LABEL:newvolumelabel`

Permits you to change the volume label. Volume labels can be as many as 12 characters. See the discussion of volume labels under MOUNT (Section 5.7) for more information about volume labels and which characters can be included.

`/MAXIMUM_FILES:n`

Specifies the maximum number of files permitted on the volume. This value must be greater than the maximum value on the current volume. See the discussion of `/MAXIMUM_FILES` under MOUNT for more detailed information on the maximum number of files permitted on various media.

`/OWNER:[g,m]`

Specifies the owner of the volume. See Section 5.1.4 for more information.

`/PROTECTION:code`

Specifies the volume protection for the volume. See Section 4.1.6 for more information on volume protection.

`/[NO]SHOW`

Specifies that the information about the volume whose home block is being updated is to be displayed on your terminal. `/NOSHOW` is the default.

Here is the display format for disks and other random-addressable devices:

```

Volume Information
Class:          Files-11 or Foreign
Device:         ddnn:
Volume label:   label
Pack Serial:    serial number supplied by manufacturer
Owner:          [g,m]
Protection:     code
Default:        code
Processor:      acpname

```

The owner field indicates the owner of the volume. The UIC must be matched for protection checks for owner privileges. The owner UIC can be changed with the `/OWNER` qualifier. See Sections 5.1.3 and 4.1.6 for more information. Protection indicates volume protection; Default indicates file protection.

`/WINDOW:n`

Specifies the default number of mapping pointers to be allocated for file windows when the file is accessed. A file window consists of a number of mapping pointers stored in memory when the file is opened. Legal values for `n` range from 1 through 127.

INITIALIZE/UPDATE (Cont.)**Notes**

INITIALIZE/UPDATE invokes the HOME utility. See the RSX-11M/M-PLUS MCR Operations Manual.

You should not use INITIALIZE/UPDATE without a good understanding of the Files-11 disk volume structure and a clear idea of the use the system makes of the values in the volume home block. For more information, see the RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual.

Error Messages

INITIALIZE/UPDATE shares many error messages with INITIALIZE.

<bell>***ATTENTION***<bell>

This volume is operating with a replacement home block.
Please take the action outlined in the documentation.

Explanation: The INITIALIZE/UPDATE command could not ensure that a good home block will be written on the volume, so it has placed an alternate home block on the first free block on the MOUNT search path. The new home block is not allocated to or mapped by any file.

User Action: Copy the volume using BACKUP (BRU) or DSC before initiating any further operations. Otherwise, the replacement home block could be destroyed, making the data on the volume unrecoverable.

INI -- Failed to find home block

Explanation: The INITIALIZE/UPDATE command could not locate the home block of the volume.

User Action: Reenter the command, including /SHOW to determine whether it is a foreign volume or not. See your system manager for further help.

INI -- Failed to place alternate home block

Explanation: When the INITIALIZE/UPDATE command cannot ensure that a good home block will be written back to the volume, it attempts to place an alternate home block on the first free block of the MOUNT search path. This message indicates the attempt failed.

User Action: The message may indicate a hardware problem, most commonly that the device is write-locked. See your system manager.

INI -- MXF less than or equal to the existing value

Explanation: The value specified in the /MAXIMUM_FILES qualifier was less than or equal to the value already specified for the volume.

User Action: Reenter the command specifying a /MAXIMUM_FILES value larger than the current one.

BACKUP**5.10 BACKUP**

BACKUP backs up and restores Files-11 volumes. It transfers files from a volume to a backup volume and retrieves files from the backup volume.

The data is directed to a volume (or volumes) during a backup operation. More than one backup set can be contained on a tape or disk, or a backup set can extend over several tapes or disks.

Format

```
BACKUP[/qualifiers]
From? sourceddnn:[filespec[s]]
To? destinationddnn:

BACKUP[/qualifiers] sourceddnn:[filespec[s]] destinationddnn:
```

Command Qualifiers**Group 1: Selective Backup and Restore**

```
/CREATED:BEFORE:(dd-mmm-yy nn:mm)
      AFTER:(dd-mmm-yy nn:mm)
/EXCLUDE
/IMAGE:arg
      SAVE
      RESTORE
/NEW_VERSION
/MODIFIED:BEFORE:(dd-mmm-yy nn:mm)
      AFTER:(dd-mmm-yy nn:mm)
/[NO]REPLACE
```

Group 2: Initialization

```
/INITIALIZE
/BAD_BLOCKS:arg
      AUTOMATIC
      MANUAL
      OVERRIDE
/ACCESSED:n
/EXTENSION:n
/HEADERS:n
/MAXIMUM_FILES:n
/INDEX:arg
      BEGINNING
      MIDDLE
      END
      n
/FILE_PROTECTION:code
/WINDOWS:n
/SAVE_SET:name
```

Group 3: Tape Control

```
/APPEND
/DENSITY:arg
      800
      1600
```

BACKUP (Cont.)

```

/ERROR LIMIT:n
/LENGTH:n
/REWIND
/LABEL:TAPE:fileset-ID

```

Group 4: Verification

```

/COMPARE
/VERIFY

```

Group 5: Display

```

/[NO]LOG
/LIST

```

Group 6: Disk Processing

```

/APPEND
/DIRECTORY
/LABEL:INPUT:volumelabel
/NOINITIALIZE
/MOUNTED
/[NO]PRESERVE
/LABEL:OUTPUT:volumelabel

```

Parameters**sourceddnn:**

Specifies the name of the device on which the input volume containing the data to be transferred has been placed.

For tapes and disks, if you are using the /IMAGE qualifier, you may specify in the following format, more than one input device for tapes and disks:

```
$ BACKUP MM0:, MM1:, MM2: (RET)
```

or

```
$ BACKUP DUO:, DU1:, DU2: (RET)
```

You can specify as many as eight drives on each command, but you can specify a particular tape or disk drive only once.

destinationdnn:

Specifies the name of the device on which the output volume to which data is being transferred has been placed.

For tapes and disks (if you are using the /IMAGE qualifier), you can specify more than one output device separated by commas, as described above.

filespec

Indicates the file specification used to select particular files or categories of files to be backed up or restored.

BACKUP (Cont.)

A file specification takes the form:

```
[g,m]filename.typ;version
```

You can specify as many as 16 file specifications per command.

Files can be backed up or restored selectively by UFD, file name, file type, or version number. Note, however, that BACKUP does not accept 0 or -1 as version numbers.

When you enter a command without a file specification, all the files on the input volume are copied to the output volume.

Any file specification field that is not specified is taken as a wildcard. If you do not specify version number, all versions will be assumed; if you specify only type, all files of all names and all versions of that type will be assumed; and so forth.

Command Qualifiers

Unless otherwise noted, the defaults for these qualifiers are the same as the equivalent attributes of the input volume.

BACKUP provides six groups of command qualifiers:

- Selective Backup and Restore
- Disk Initialization
- Tape and Disk Control
- Verification
- Display
- Disk Processing

Group 1: Selective Backup and Restore

```
/CREATED:BEFORE:arg
      AFTER:arg
```

Specifies selection of files created before or after the date and time given by arg.

The required date/time argument can be in several formats. You must specify either a date or a time or both. If you do not specify a date, the default is today. If you do not specify a time, the default is now. Here are some examples:

```
/BEFORE:28-MAR-85
/AFTER:14:00
/BEFORE:(1-APR-85 12:00)
/AFTER:(1-APR-85 12:00)
/BEFORE:4/1/85
/AFTER:(4/1/85 12:30)
```

Note that if you wish to enter both a date and a time, you must use the parentheses. These formats are used wherever you specify a date or a time or both in a BACKUP command.

BACKUP (Cont.)**/EXCLUDE**

Specifies that all files specified with the source device are to be excluded from the backup or restore operation, instead of included in it. For example:

```
$ BACKUP/EXCLUDE DB0:[306,6]*.* MM0: (RET)
```

This command transfers all files except those in the directory [306,6].

/IMAGE:arg

SAVE

RESTORE

Specifies that you want to do a multiple disk-to-disk backup or restore operation.

If you want to do a backup operation, you must use /IMAGE:SAVE when you create the backup file that represents the image copy of the input disk or disks. For example, this qualifier must be used when you copy a large disk to several small disks, or if you copy several small disks to a mounted large disk.

If you are doing a restore operation, you must use /IMAGE:RESTORE if you are restoring from a backup file that represents the image copy of the original disk.

NOTE

You cannot use the /IMAGE qualifier when backing up or restoring from one disk or diskette to another disk or diskette. For one-to-one backup and restore operations, use the /INITIALIZE qualifier.

/NEW_VERSION

Specifies that a new version of the file be created in the event of conflicts that occur during restore operations and during backups to a mounted disk when using the /IMAGE:SAVE qualifier. Compare with /REPLACE.

/MODIFIED:BEFORE:arg

AFTER:arg

Specifies selection of files modified before or after the specified date or time, or both.

The required date/time argument can be in several formats. You must specify either a date or a time or both. If you do not specify a date, the default is today. If you do not specify a time, the default is now. Here are some examples:

```
/BEFORE:28-MAR-85
/AFTER:14:00
/BEFORE;(1-APR-85 12:00)
/AFTER:(1-APR-85 12:00)
/BEFORE:4/1/85
/AFTER:(4/1/85 12:30)
```

BACKUP (Cont.)

Note that if you wish to enter both a date and a time, you must use the parentheses. These formats are used wherever you specify a date or a time or both in a BACKUP command.

/[NO]REPLACE

Specifies whether or not files from the input volume should replace files having identical file specifications on the output volume.

A conflict occurs when a file on the input volume and a file on the output volume have identical file specifications. The default qualifier /NOREPLACE resolves that conflict by keeping the file on the output volume rather than replacing it with that on the input volume. A message informs you that the file has not been superseded. /REPLACE resolves the conflict by deleting the file on the output volume and replacing it with the file being restored from the input volume.

Group 2: Initialization**/INITIALIZE**

Specifies that you wish to initialize the output disk before proceeding with the data transfer. You must specify this qualifier if you wish to use any of the other initialization qualifiers discussed here. If you do not specify this qualifier, but specify other initialization qualifiers, you will be asked by a prompt whether you wish to initialize before proceeding or not. The default is /NOINITIALIZE.

See Section 5.9 for a description of INITIALIZE and its qualifiers.

/BAD_BLOCKS:arg
AUTOMATIC
MANUAL
OVERRIDE

Enters the locations of bad blocks on the volume being initialized. See the ANALYZE/MEDIA command description for more information on finding and marking bad blocks.

AUTOMATIC specifies that bad-block information for the volume be determined automatically from the file BADBLK.SYS, created by the BAD utility. This is the default.

MANUAL specifies that bad-block information for the volume is to be entered manually.

OVERRIDE applies only to last-track devices. It makes the disk appear not to have bad-block information written in the last track, even if this information is actually present.

/ACCESSED:n

Specifies the default number of File Control Blocks per volume to be kept in memory by the ACP. See the description of the /ACCESSED qualifier to MOUNT (Section 5.7) for more information.

BACKUP (Cont.)**/EXTENSION:n**

Specifies the number of blocks by which the file will be extended when its allotted space is exhausted.

/HEADERS:n

Specifies the number of file headers to be allocated initially in the index file.

The five system files (INDEXF.SYS, BITMAP.SYS, CORIMG.SYS, BADBLK.SYS, and 000000.DIR) are not included in the value for n.

The default value for n is proportional to the capacity of the magnetic medium. See the description of /HEADERS in Section 5.9 for information on the different magnetic media.

/MAXIMUM_FILES:n

Specifies the maximum number of files permitted on the volume.

The value of n includes the five system files. The value of n must be less than or equal to the maximum value of n given in the description of /MAXIMUM FILES in Section 5.7. The description also includes the default value of n for each device.

/INDEX:arg

BEGINNING
MIDDLE
END
n

Specifies the location of the index file on the volume. This qualifier forces a specified location for the index file, the Master File Directory, and the storage allocation file. This is usually done to minimize access time.

Four possibilities are available for the argument field:

BEGINNING	The beginning of the volume. This is the default for tapes and DECTapes.
MIDDLE	The middle of the volume. This is the default for disks.
END	The end of the volume.
n	Logical block n.

The default position is the position of the file on the input volume.

/FILE_PROTECTION:(code)

Specifies the default protection for all files created on the volume being initialized. See Section 4.1.6 for more information on file protection.

BACKUP (Cont.)

The default system protection code is as follows:

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

The default protection code is the protection code of the input volume.

/WINDOWS:n

Specifies the number of mapping pointers to be allocated for file windows. A file window consists of a number of mapping pointers stored in memory when the file is opened.

The default is the number of mapping pointers on the input volume.

See the discussion of the /WINDOWS qualifier in Section 5.7 for more information.

/SAVE_SET:name

Specifies the name of the backup set to be placed on tape, disk, or diskette. This name can be up to 12 characters long.

When you place several backup sets on a tape, disk, or diskette output volume, each backup set should have a different backup set name in case you need to restore your backup sets. If the backup sets have the same backup set name, only the first backup set created can be restored.

If you do not use this qualifier, the backup set name for a tape or an unmounted disk or a diskette defaults to the name of the volume being backed up. If you do not use this qualifier when performing a multivolume backup operation to a mounted output disk or diskette, the backup set name defaults to [0,0]BACKUP.SYS.

When you use /SAVE_SET during a tape restore operation, BACKUP searches the first tape for the specified backup set name. If you do not specify a backup set name during a tape restore operation, the first backup set on the tape is backed up. You can restore sequential backup sets from a tape without having to rewind the tape between restore operations. Note that BACKUP does not rewind the first tape in a backup set unless you specify the /REWIND qualifier.

When you use /SAVE_SET during a disk or diskette restore operation, BACKUP searches the first disk or diskette for each backup set you specify. Each backup set is then restored in the order you specified.

Note that all backup sets must begin on the first tape, disk, or diskette in your set. For example, if you have six backup sets on tape, diskette, or disk, the first five backup sets must begin and end on the first reel of tape or the first disk or diskette. The sixth backup set must also begin on the first reel of tape, or the first disk or diskette, but it may span more than one tape, disk, or diskette.

BACKUP (Cont.)**Group 3: Tape Control****/APPEND**

Appends new data to a tape that already contains one or more backup sets.

If the tape is not positioned at the beginning, /APPEND causes output to be written where the tape is currently positioned. If you want to start writing at logical end-of-tape, you must specify /APPEND/REWIND, which rewinds the tape and then spaces forward to the end of the volume.

/DENSITY:arg
800
1600

Specifies the density in bits per inch (bpi) for the medium loaded on the device. If you are adding your tape drives that allow other densities, you must use this qualifier. See Section 5.7 for a list of acceptable densities for DIGITAL devices.

The default is 800 bpi.

/ERROR_LIMIT:n

Specifies the number of nonfatal I/O errors that the Backup and Restore Utility tolerates on tape reads before terminating execution.

The default value for n is 25.

/LENGTH:n

Specifies the length of the output tape in decimal feet. You may want to use this qualifier when you know that the last part of a tape is defective and must not be used. You can specify a shorter length and ensure that you do not write on that part of the tape.

The default length is the physical length of the output tape.

/REWIND

Rewinds the first tape of a tape set before executing the command line.

See /APPEND for further information on /REWIND.

/LABEL:TAPE:fileset-ID

Specifies a 6-character ANSI fileset-ID. On output operations this ID is written on the tape. On input operations the ID is compared with the ID already on the input tape to ensure that you have mounted the correct tape.

Group 4: Verification

The following qualifiers allow you to verify the accuracy of data transfers.

BACKUP (Cont.)

/COMPARE

Compares the data on the output volume to the data on the input volume and reports any differences. /COMPARE does not copy any data. See /VERIFY.

You must compare equivalent volumes. You can assure that the volumes are equivalent by including the /COMPARE qualifier in a command that is otherwise identical to the BACKUP command that was entered when the output volume was created.

The restriction does not apply to the /[NO]INITIALIZE and /COMPARE qualifiers.

/VERIFY

Copies data from the input volume to the output volume, then performs a compare operation, and reports any differences.

Group 5: Display

The following qualifiers display information about the data transferred.

/[NO]LOG

Displays at your terminal the UFD and file name of each file being backed up or restored. /NOLOG is the default.

/LIST

Displays at your terminal the backup set names or files on the specified tape or disk volume. In a multivolume tape set, the directory is on the first tape of a set. In a multivolume disk set, the directory is on the first disk of the set. See the example.

Group 6: Disk Processing

The following qualifiers allow you to control the way the operating system manages disk data transfer operations.

/APPEND

Appends new data to a disk or diskette that already contains one or more backup sets.

When you are using the /IMAGE qualifier, /APPEND causes BACKUP to add the new backup set at the end of the last backup set on your disk or diskette.

You may also append a new backup set to a tape that already contains one or more backup sets. See Group 3, which describes tape control.

You cannot use the /APPEND qualifier during a backup operation to a mounted disk or diskette.

BACKUP (Cont.)**/DIRECTORY**

Directs BACKUP to create directories (if they do not already exist) on a mounted output volume, then copy into the directories the files from the same directory on the input volume. If you do not specify /DIRECTORY, BACKUP does not copy the file. /DIRECTORY is used only with the /NOINITIALIZE qualifier.

/LABEL:INPUT:volumelabel

Specifies the volume label of the input disk. This is not a required parameter; if you specify it, however, BACKUP checks the volume label.

/NOINITIALIZE

Specifies that you do not want to initialize the output disk, as it already contains a Files-11 structure and is mounted as a Files-11 volume.

/MOUNTED

Allows you to back up files from a disk that is mounted as Files-11. Transfers are faster from an unmounted volume.

/[NO]PRESERVE

Specifies whether you want BACKUP to preserve file identifiers during a transfer using disks or diskettes. The default is /PRESERVE.

/LABEL:OUTPUT:volumelabel

Specifies a volume label for the output disk. The default label is the label of the input volume.

Examples

```
$ BACKUP/LABEL:INPUT:RANDOM/SAVE_SET:19JUL85 (RET)
From? DK0: (RET)
To: MM0: (RET)
```

BAC - Starting tape 1 on MM0:

BAC - End of tape 1 on MM0:

BAC - Completed

This example transfers all files on the RK05 disk labeled RANDOM to the tape mounted on MM0:. The backup set name 19JUL85 identifies the data written on the output tape.

```
$ BACKUP/REWIND/SAVE_SET:18MAY85/NOINITIALIZE MM0: DM0: (RET)
```

BACKUP (Cont.)

This example rewinds the tape on drive MM0:, then locates the backup set with the name 18MAY85 (there may be more than one backup set on a tape) and transfers all files in the backup set to DM0: which is mounted and has a Files-11 structure. The /NOINITIALIZE qualifier instructs BACKUP not to initialize DM0:.

```
$ BACKUP/LIST MM0: (RET)

VOL1 18MAY85 INPUT1 18-MAY-85
VOL1 BACKUP2 INPUT2 19-APR-85
VOL1 19APR85 RANDOM 19-APR-85
```

This example shows a directory of a tape volume. It displays the tape volume number, VOL1, the names of three volumes, and three backup sets contained on the tape.

BACKUP can also be used to display the names of the files in a given backup set by specifying the backup set name along with the /LIST qualifier, as in the following example:

```
$ BACKUP/SAVE SET:19APR85/LIST MM0: (RET)
VOL1 19APR85 RANDOM 19-APR-85 23:37:11
[303,13]
27DECE.LST;1
ASTCRSH.MAC;2
APNDXC.TXT;1
X.MAC;1
X.OBJ;1
X.TSK;1
X.MAP;1
[306,6]
27DECE.CDA;1
RSX11M.STB;3
```

BAC - Completed on MM0:

Notes

The BACKUP command invokes BRU, the Backup and Restore Utility. For more information on BRU, see the RSX-11M/M-PLUS Utilities Manual.

You must specify either /INITIALIZE or /NOINITIALIZE. If you specify neither, BACKUP will ask if you wish to initialize the output device.

BACKUP is not a privileged command, but you should not use it if you are not familiar with BRU.

Error Messages

For many nonfatal errors, BACKUP continues with the operation, but prints a warning message such as the following:

```
BAC -- *WARNING* APPENDING AT 800 BPI

BAC -- *WARNING* BOOT BLOCK IS BAD
```


BACKUP (Cont.)

BAC -- *WARNING* CANNOT RESTORE CONTIGUOUSLY

BAC -- *WARNING* EOT MARKER ERROR

BAC -- *WARNING* FILE IDS WILL NOT BE PRESERVED

BAC -- *WARNING* NONFATAL QUALIFIER CONFLICTS BEING IGNORED

These and other warning messages are explained in the RSX-11M/M-PLUS Utilities Manual.

BACKUP also issues purely informational messages such as the following:

Completed

Starting tape n on ddnn:

These messages require no response.

BACKUP requests response from the user with messages such as the following:

Please answer Yes or No

Mount another tape

The following are the most common error messages from BACKUP. Other error messages are described in the RSX-11M/M-PLUS Utilities Manual.

BAC -- *FATAL*-Allocation for sys file exceeds volume limit

Explanation: One of the five files created when a volume is initialized does not fit on the output disk. This message usually occurs when the output disk is smaller than the input disk.

User Action: Reinitialize the disk using the /INDEX:BEGINNING qualifier to BACKUP or INITIALIZE to force the allocation to start at the beginning of the disk. You can also use the /MAXIMUM FILES or /HEADERS qualifiers to reduce the size of INDEXF.SYS.

BAC -- *FATAL*-Attach failed on ddnn:

Explanation: BRU could not attach the specified device.

User Action: Check to see if another task has the device attached, or if the device has a volume mounted on it.

BAC -- *FATAL*-Device conflict

Explanation: Command specified both a tape and a disk drive as part of the input or output device specification.

User Action: You can specify either a disk or a tape but not both. Check command for proper device names and reenter the command.

BACKUP (Cont.)

BAC -- *FATAL*-Device not in system

Explanation: Command specified a device that is not part of the current system.

User Action: Check command for proper device names and reenter.

BAC -- *FATAL*-Device not supported

Explanation: Command specified a device that is not a tape or disk.

User Action: Check command for proper device names and reenter.

BAC -- *FATAL*-Disk read error

Explanation: An unrecoverable read error occurred on the output disk. This error could be caused by an undetected bad block.

User Action: Run BAD on the disk and then use BACKUP with the /BAD_BLOCKS:AUTOMATIC qualifier.

BAC -- *FATAL*-Disk write error

Explanation: An unrecoverable write error occurred on the output disk. This error could be caused by an undetected bad block.

User Action: Run BAD on the disk and then use BACKUP with the /BAD_BLOCKS:AUTOMATIC qualifier.

BAC-- *FATAL*-End of volume encountered. Backup set not found

Explanation: The backup set specified for a restore operation is not on the tape volume.

User Action: Mount the correct tape volume and retry the operation.

5.11 SET AND SHOW

You can display assignments and set and display certain device characteristics.

SHOW ASSIGNMENTS

5.11.1 SHOW ASSIGNMENTS

SHOW ASSIGNMENTS displays at your terminal all local and login logical-device assignments. Privileged users can display assignments from other terminals and global assignments.

Logical-device assignments are established by ASSIGN, by SET DEFAULT, and by ACNT, the Account File Maintenance Program.

Format

```
SHOW ASSIGNMENTS[/qualifier]
```

Command Qualifiers

```
/LOCAL
/GLOBAL
/TERMINAL:ttnn:
/LOGIN
/SYSTEM (synonym for /GLOBAL)
```

Parameters

None.

Command Qualifiers

/LOCAL

Specifies that local and login logical assignments from your terminal are to be displayed on your terminal. This is the default.

/GLOBAL

Specifies that all logical assignments in the system, including global assignments, are to be displayed on TI:. This is a privileged qualifier.

/TERMINAL:ttnn:

Specifies that local and login assignments from terminal ttnn: are to be displayed on TI:. This is a privileged qualifier.

/LOGIN

Specifies that local and login logical assignments from your terminal are to be displayed on your terminal. This is a privileged qualifier.

Examples

```
$ SHOW (RET)
Function? ASSIGNMENTS (RET)
LP0:    DR1:    LOCAL  TI - TT11:
TP0:    TT11:   LOCAL  TI - TT11:
SY0:    DB1:    LOGIN  TI - TT11:
```

SHOW ASSIGNMENTS (Cont.)

This example shows the display for a user who has used ASSIGN to assign local logical names to a number of devices. These logical names can be used in place of the device names. The logical name is displayed on the left and the physical device name (or pseudo device name) is on the right. The type of assignment -- LOCAL or LOGIN -- is given, and the TI:. Output to LP0: will actually go to DR1:. In this case, the logical name takes the place of a physical device name. The user cannot direct output to physical LP0:. Output to TP0: will go to TT11:, which is TI:. In addition, the user has the login assignment of SY0: to DB1:. This means that DB1: is the default device for this user.

\$ SHOW ASSIGNMENTS/LOCAL (RET)

This example is the equivalent of the previous example.

\$ SHOW ASSIGNMENTS/TERMINAL:TT37: (RET)

MP0: SY0: LOCAL TI - TT37:
SY0: SY0: LOGIN TI - TT37:

This example, which requires a privileged terminal, displays the logical assignments from terminal TT37:. This user has given the local logical name MP0: to the pseudo device, SY0:. The login assignment gives the login logical name SY0: to the pseudo device SY0:. SY0: is the name of the default device.

\$ SHOW ASSIGNMENTS/GLOBAL (RET)

SY0: DB1: LOCAL TI - TT42:
SY0: DB1: LOGIN TI - VT4:
SY0: SY0: LOGIN TI - TT4:
SY0: SY0: LOGIN TI - TT15:
.
.
.
WK0: LB0: GLOBAL
IN0: SY0: GLOBAL
EX0: SY0: GLOBAL

This example, which requires a privileged terminal, displays all logical assignments for the entire system.

SET DEVICE

5.11.2 SET DEVICE

SET DEVICE establishes certain device attributes. With the exception of SET DEVICE:TI:/WIDTH:n, this is a privileged command.

Format

```
SET DEVICE
Device? ddnn:
Attribute? attribute

SET DEVICE:ddnn:/attribute
```

Device Attributes

```
/[NO]CHECKPOINT_FILE:n
/[NO]LOWERCASE
/[NO]PUBLIC
/[NO]SYSTEM
/WIDTH:n
```

Parameter

DEVICE:ddnn:

Specifies the device for which attributes are to be set. Note that, 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 Attributes

/[NO]CHECKPOINT_FILE:n

Sets aside n (decimal) blocks on the specified volume in [0,0]CORIMG.SYS, the checkpoint file. The volume must be in Files-11 format. Only one checkpoint file is permitted on each volume, but several volumes may have checkpoint files. The system searches for checkpoint files in the order they were created. If space is not available in the first checkpoint file created (the primary file), the system searches in the second file to be created, and so on. Checkpoint files can be allocated on volumes mounted on private devices. In general, the system should have checkpoint space equal to twice the total amount of memory that all running tasks are using.

The /NOCHECKPOINT_FILE attribute makes the checkpoint file unavailable. If the specified file contains checkpointed tasks, the file is not deactivated until the tasks have returned to memory, but no new tasks will be checkpointed to the file. The checkpoint file is still on the volume but is zero blocks long.

SET DEVICE (Cont.)

/[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.

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

/[NO]PUBLIC
/[NO]SYSTEM

The /PUBLIC attribute sets the device as a public device. This command also mounts the device. Public devices are accessible to all users.

The /NOPUBLIC attribute removes the public status of a device. This is the default setting.

/[NO]SYSTEM is a synonym included for VAX/VMS compatibility.

Section 5.1.3 contains a discussion of public and private devices.

/WIDTH:n

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

For line printers, n must be greater than 0 and not greater than 255. The line-printer driver discards all characters in a record (line) that do not fit in the I/O buffer.

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.

Nonprivileged users can set the width of their terminals using this command or SET TERMINAL/WIDTH:n.

Examples

```
$ SET (RET)
Function? DEVICE (RET)
Device? LPl:(RET)
Attribute? WIDTH:80 (RET)
```

This example sets the I/O buffer size (line length) to 80 (decimal) characters on the line printer LPl:.

SET DEVICE (Cont.)

```
$ SET DEVICE:LP1:/LOWERCASE (RET)
```

This example sets LP1: so that all lowercase characters sent to it will be printed in lowercase. The command must be issued even if the printer is controlled by the Queue Manager and has been initialized lowercase.

```
$ SET DEVICE:DB3:/CHECKPOINT_FILE:256 (RET)
```

This example creates a checkpoint file of 256 (decimal) blocks on DB3:.

Notes

SHOW DEVICES displays the attributes set by SET DEVICE.

Error Messages

SET -- Feature not supported

Explanation: The command specified an optional feature that was not incorporated into the system at system generation.

User Action: See your system manager.

SET -- Pseudo device error

Explanation: Command attempted to set a pseudo device public.

User Action: A pseudo device cannot be set public. Retype command using physical device name.

SET -- Write check not supported

Explanation: Command attempted to enable write-checking on a device that does not support it.

User Action: Write-checking is not supported on magnetic tapes, RL0ls, or RX0ls.

SHOW DEVICES**5.11.3 SHOW DEVICES**

SHOW DEVICES displays information about the devices included in the system.

Format

```
SHOW DEVICES[/attribute] [dd[nn:]]
```

Device Attributes

```
/[NO]PUBLIC
/WIDTH
/[NO]SYSTEM (synonym for /PUBLIC)
```

Parameter

dd[nn]:

Specifies the device or devices of a particular type on the system for which attributes are to be displayed. dd: is the 2-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.

The command SHOW DEVICES dd[nn:] is useful when you plan to allocate a drive, as it shows you which drives are available. Since the same controller may control devices whose disks are not interchangeable, a generic allocate (ALLOCATE DB) may allocate a device which cannot accept your disk. See examples.

Device Attributes

If you do not include an attribute, SHOW DEVICES displays a list of all the devices on the system, including terminals and pseudo devices.

```
/[NO]PUBLIC
/[NO]SYSTEM
```

Displays a list of all devices set (or not set) PUBLIC. /[NO]SYSTEM is a synonym included for VAX/VMS compatibility.

```
/WIDTH
```

Displays the size of the I/O buffer (line length) for a particular device, including a terminal. If no device is specified, /WIDTH defaults to TI:.

Display Information

The display from SHOW DEVICES can include a number of messages.

ddnn:

A device name in the first column indicates the device or pseudo device for which information is being displayed. A device name in the second column indicates a device to which the corresponding device in the first column has been redirected (logically assigned).

SHOW DEVICES (Cont.)

TTnn:

A terminal name in the second column, followed by the word PRIVATE, indicates that the device named in the first column has been allocated by the user logged on to the terminal in the second column.

MOUNTED

Indicates that the device is mounted. For privileged users, the message also includes the volume label.

BUF=

Indicates the line length (I/O buffer size).

PUBLIC

Indicates that the device has been set public.

If your command was SHOW DEVICES /PUBLIC, or /NOPUBLIC, the display is PUB=ddnn: or NOPUB=ddnn:

TYPE=

Indicates the device type by model name, for example, RL01, RX02. If you wish to display devices by type, use the 2-letter device mnemonic.

MARKED FOR DISMOUNT

Indicates that a mounted device has been marked for dismount, but that the dismount has not been completed, meaning that files are still open on the volume. The volume can be remounted while it is marked for dismount.

OFFLINE

Indicates that the system tables contain entries for the device, but that it is not included in the current configuration.

[directory] LOGGED ON

Indicates that the user identified by [directory] is logged in on the terminal.

LOADED

Indicates that a loadable device driver is currently loaded.

UNLOADED

Indicates that a loadable device driver is currently not loaded.

SPOOLED

Indicates that a device is under the control of the Queue Manager.

SHOW DEVICES (Cont.)

WCHK=
NOWCHK=

Indicates a device with write-checking enabled or not enabled.

Examples

```

$ SHOW DEVICES (RET)
DB0: PUBLIC MOUNTED LOADED TYPE=RP06
DB1: PUBLIC MOUNTED LOADED TYPE=RP06
DB2: PUBLIC LOADED TYPE=RP04
DB3: PUBLIC MOUNTED LOADED TYPE=RP06
DR0: VT3: - PRIVATE MOUNTED LOADED TYPE=RM03
DR1: LOADED TYPE=RM03
DK0: LOADED
LP0: DB0: SPOOLED LOADED
LP1: DB0: SPOOLED LOADED
CO0: TT0:
TT0: LOADED
TT1: [7,40] [1,1] - LOGGED ON LOADED
TT2: [7,40] [14,10] - LOGGED ON LOADED
TT3: [303,5] [303,5] - LOGGED ON LOADED
      .
      .
      .
TT66: LOADED
TT67: LOADED
NL0: LOADED
VT0: LOADED
VT1: LOADED
VT2: [1,1] [1,1] - LOGGED ON LOADED
VT4: [303,5] [303,5] - LOGGED ON LOADED
TI0:
CL0: TT0:
SP0: DB0:
LB0: DB0:
SY0: DB0:

```

This example shows the display from SHOW DEVICES. All devices and pseudo devices are included. For terminals, the first [g,m] field is the login UIC and the second is the default UFD. The only information shown for pseudo devices is the name and the device to which they are redirected. Privileged users also see volume labels.

```

$ SHOW DEVICES DB: (RET)
DB0: PUBLIC MOUNTED LOADED TYPE=RP06
DB1: PUBLIC MOUNTED LOADED TYPE=RP06
DB2: PUBLIC LOADED TYPE=RP04
DB3: PUBLIC MOUNTED LOADED TYPE=RP06

```

This example displays information about all devices of the type DB:. The mnemonic (DBnn:) identifies the device controller and the model name (RP04) indicates the physical device type. RP04 disks do not fit RP06 drives, even though the device controller is the same.

SHOW DEVICES (Cont.)

```
$ SHOW DEVICES/PUBLIC (RET)
PUB=DB0:
PUB=DB1:
PUB=DB2:
PUB=DB3:
PUB=DS0:
PUB=DS1:
PUB=LP0:
PUB=LP1:
```

This example displays information about all public devices. Public devices are accessible to all users. They may be mounted by any user to assure continued access to the volume mounted on the device.

```
$ SHOW DEVICE/WIDTH LP0: (RET)
BUF=LP0:00132.
$ SHOW DEVICE/WIDTH TT11: (RET)
BUF=TT11:00080.
```

This example displays the line lengths of a line printer and a terminal. The line length is determined by the size of the I/O buffer, which is established through SET DEVICE:ddnn:WIDTH.

CHAPTER 6
PROGRAM DEVELOPMENT

6.1 INTRODUCTION

Program development on RSX-11M operating systems is accomplished in four steps:

1. Creating a program in a supported source language using an editor
2. Compiling or assembling the source program to create a binary object file
3. Turning the object file into a runnable task image using the Task Builder
4. Running the program as a task on the system

If you are not familiar with the rudiments of this process, see the Introduction to RSX-11M.

To create a source program, you must be familiar with the file-creation facilities of the system discussed in Chapter 4 of this manual. You will also need knowledge of one of the programming languages available separately as software options. Commands described in this chapter on program development include commands to maintain libraries used by programs, and the LINK command, which controls the operation of the Task Builder.

RSX-11M systems offer extensive program development facilities. These facilities are introduced in the RSX-11M/M-PLUS Guide to Program Development. This manual introduces the most important software tools used in program development and takes you through the program development process in considerable detail.

LINK

6.2 LINKING THE TASK

The MACRO-11 Relocatable Assembler or high-level language compiler translates the program you wrote (the source program file) into a binary object file. You must then link or task build the object file. Linking or task building makes a runnable task out of one or more object files plus object modules from user-written or system library files.

6.2.1 Introduction to the LINK Command

The LINK command invokes the Task Builder, a system utility that builds the tasks, formally called task images, that run on the system.

The unadorned LINK command includes a number of defaults that produce a standard task image file, having the file type .TSK. The Task Builder (TKB), however, provides you with many choices that are not included in the defaults. You can use qualifiers to the LINK command to perform the following functions:

- Create TKB output files in addition to the task image file
- Identify certain kinds of unusual tasks
- Use hardware options that may not be present on all systems
- Control the nature of the task build itself

In addition to the LINK command qualifiers, there are also Task Builder options that permit you to specify characteristics of the task you are building.

The defaults provide you with a simple means of building a standard task. You specify in the LINK command line and, therefore, pass to the Task Builder an object file (default file type .OBJ) produced by the MACRO-11 Relocatable Assembler or one of the high-level language compilers. The Task Builder then produces a runnable task image. Invoking the Task Builder is done with a command such as the following:

```
$ LINK (RET)
File(s)? SAMPLE (RET)
$
```

This command produces a single output file called SAMPLE.TSK. The return of the implicit prompt tells you that the task build completed successfully.

If, in addition to the task image file, you should want a Task Builder map file showing the size and location of parts of the task, issue a command such as the following:

```
$ LINK/MAP SAMPLE (RET)
$
```

This produces two output files, SAMPLE.TSK and SAMPLE.MAP.

The Task Builder spools SAMPLE.MAP to the line printer and puts SAMPLE.MAP in your directory.

LINK (Cont.)

The following command also requests that a map file be created:

```
$ LINK SAMPLE.OBJ/MAP (RET)
$
```

This produces two output files, SAMPLE.TSK and SAMPLE.MAP. The Task Builder puts SAMPLE.MAP in your directory, but does not spool it to the line printer.

If you are linking more than one object module to form a task, the Task Builder, by default, names the output files after the first task named in the command line. For example, the following command:

```
$ LINK MOE, CURLY, LARRY (RET)
$
```

produces a task image file named MOE.TSK. That is, LINK uses the name of the first input file, MOE, as the name of the task unless you change the name of the task. Use the /TASK qualifier to change the name of the task image file. For instance, the following command:

```
$ LINK/TASK:SHEMP MOE, CURLY, LARRY (RET)
$
```

produces a task image file named SHEMP.TSK. Or, use the following command:

```
$ LINK MOE, CURLY/TASK, LARRY (RET)
$
```

to produce a task image file named CURLY.TSK.

Note, however, that the following LINK command:

```
$ LINK/MAP MOE, CURLY/TASK, LARRY (RET)
$
```

produces a task image file named CURLY.TSK and a map file named MOE.MAP because MOE is the first input file. If you use the following form:

```
$ LINK/MAP:SAM MOE,CURLY,LARRY (RET)
```

the map file is named SAM instead of MOE.

If, in addition, you want the task build to abort after the Task Builder finds three errors, issue a command such as this:

```
$ LINK/MAP/ERROR_LIMIT:3 SPARKY (RET)
$
```

As before, this produces two output files, the task image named SPARKY.TSK and the map named SPARKY.MAP. In addition, if the Task Builder finds three errors, it sends a message to your terminal and aborts the task build.

LINK (Cont.)

Finally, there are Task Builder options. If you specify the /OPTION qualifier in your LINK command, LINK prompts you for further input. For example:

```
$ LINK/MAP/OPTIONS BIGJON (RET)
Option? UNITS=7 (RET)
Option? (RET)
$
```

The UNITS option specifies the number of I/O units for the task. There are more than two dozen options you can specify. They are listed in Table 6-1.

If you want to specify options, there are three ways of doing so. You must first include the /OPTION qualifier in your LINK command. You will be prompted **Option?** Then you can do the following:

1. Specify options on separate lines. Simply enter an option with its arguments and press RETURN. You will be prompted **Option?** until you terminate the prompt with just a RETURN.
2. Specify options in a separate file. When prompted by **Option?**, simply name that file with a preceding at sign (@) and then press RETURN.
3. Specify options in a separate file and name that file as an argument to the /OPTIONS qualifier. Note, however, that you will not be prompted for options.

Here are three examples.

```
$ LINK/MAP/OPTIONS BIGJON (RET)
Option? UNITS=7 (RET)
Option? ASG=SY0:6:7 (RET)
Option? (RET)
$
```

or

```
$ LINK/MAP/OPTIONS BIGJON (RET)
Option? @SPARKY.CMD (RET)
Option? (RET)
$
```

or

```
$ LINK.MAP/OPTIONS:@SPARKY.CMD BIGJON (RET)
```

The LINK command and its qualifiers are all DCL commands, and the format is DCL format. The Task Builder, however, does not recognize DCL format. It recognizes only Task Builder format. The translation from DCL format to TKB format is performed by DCL.

Issuing the LINK command to DCL initiates the following process:

1. DCL reads the command line and checks it for correct DCL format.
2. DCL translates the command line into correct TKB format.

LINK (Cont.)

3. DCL places the TKB-format command in a TKB indirect command file and passes the indirect command file to the Task Builder.
4. The Task Builder builds the task as directed and creates all requested output files.
5. The Task Builder returns control to DCL and DCL deletes the TKB indirect command file.

If you are building standard tasks using the defaults and following the examples given in the command description (Section 6.2.2), this process will serve you well.

However, you will have to refer to the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for information on how the Task Builder responds to your LINK command. The Task Builder manual explains all the output files produced by TKB and how TKB interprets your LINK command. All the options are explained in the manual, all the terms used in the TKB environment are defined, and, in general, it elucidates the capabilities of this complex and flexible software tool.

The description of the LINK command in this manual is not intended to teach you how to use the Task Builder, but only to teach you how to use the LINK command. To learn to use the Task Builder, you must go to the Task Builder manual.

Because TKB format and DCL format are different, understanding how the DCL commands work with the Task Builder commands may be difficult at first. The TKB translation of your LINK command is available to you in two forms. First, you can use the SET DEBUG command (Section 1.2.15), which echoes on your terminal the translation of any DCL command. Second, you can use the /SAVE qualifier to LINK. The /SAVE qualifier preserves the indirect command file created by DCL to be passed to the Task Builder. This file is named ATLNK.TMP. This file is also useful if you wish to repeat a particular task build. See the discussion of /SAVE.

The following description should help you switch from DCL format to TKB format.

In TKB format, you name any output files you want produced, separated by commas, to the left of an equal sign (=). The order you name them determines which output files you receive. The Task Builder can produce up to three kinds of output files. See the following examples:

The following DCL command:

```
$ LINK SAMPLE.OBJ (RET)
```

produces the following command in TKB format in the TKB indirect command file:

```
TKB>SAMPLE.TSK=SAMPLE.OBJ (RET)
```

In this first example, the Task Builder takes SAMPLE.OBJ as its input file and produces as its only output file the task image SAMPLE.TSK. The task image file takes the first position in the list of TKB output files to the left of the equal sign (=).

LINK (Cont.)

The following DCL command:

```
$ LINK/MAP SAMPLE.OBJ (RET)
```

produces the following command in TKB format:

```
TKB>SAMPLE.TSK,SAMPLE.MAP=SAMPLE.OBJ (RET)
```

In this second example, the Task Builder takes SAMPLE.OBJ as its input file and produces two output files, SAMPLE.TSK and SAMPLE.MAP. The map file takes the second position in the list of TKB output files to the left of the equal sign.

The following DCL command:

```
$ LINK/MAP/SYMBOL_TABLE SAMPLE.OBJ (RET)
```

produces the following command in TKB format:

```
TKB>SAMPLE.TSK,SAMPLE.MAP,SAMPLE.STB=SAMPLE.OBJ (RET)
```

In this third example, the Task Builder takes SAMPLE.OBJ as its input file and produces three output files, SAMPLE.TSK, SAMPLE.MAP, and SAMPLE.STB. The symbol definition file takes the third position in the list of TKB output files to the left of the equal sign.

In TKB format, you alter the output file by adding switches (consisting of a slash and two letters) to each of these output file specifications.

For instance, the following DCL command:

```
$ LINK/CODE:FIS SAMPLE.OBJ (RET)
```

produces the following command in TKB format:

```
TKB>SAMPLE.TSK/FP=SAMPLE.OBJ (RET)
```

In each case, the Task Builder builds a task that uses the Floating Point Processor, which is optional hardware.

Qualifiers can also be used on map files. The following DCL command:

```
$ LINK/CROSS_REFERENCE SAMPLE.OBJ (RET)
```

produces the following command in TKB format:

```
TKB>SAMPLE.TSK,SAMPLE.MAP/-SP/CR=SAMPLE.OBJ (RET)
```

In this case, the Task Builder map includes a listing of symbol cross-references in the program. DCL does not require you to use the /MAP qualifier if you use another qualifier, such as /CROSS_REFERENCE, that implies that you want a map file.

In the description of the LINK command that follows, each DCL qualifier is identified with its equivalent in TKB format.

LINK (Cont.)

You can invoke the Task Builder from DCL in four ways:

1. Issuing the LINK command with DCL qualifiers
2. Invoking a DCL indirect command file including a LINK command with DCL qualifiers, such as DCL>@LINKFILE
3. Issuing the LINK command and calling a TKB indirect command file, such as DCL>LINK @TKBFILE
4. Running the Task Builder with RUN \$TKB and issuing TKB commands

The first two methods of invoking the Task Builder require the LINK command and DCL format. The latter two methods require commands in TKB format.

6.2.2 LINK

LINK invokes the Task Builder, which links object modules and routines from user and system libraries to form an executable task.

Format

```
LINK[/qualifier[s]] filespec[/parameter
qualifier[s]][,filespec[,s]]
```

Command Qualifiers

```
/ANCILLARY_PROCESSOR[:n]
/[NO]CHECKPOINT:arg
        SYSTEM
        TASK
/CODE:(arg[,s])
        EAE
        [NO]FPP
        PIC
        POSITION_INDEPENDENT
/COMPATIBLE
/CROSS REFERENCE
/DEBUG[:filespec]
/[NO]EXECUTABLE[:filespec]
/ERROR_LIMIT:n
/FAST
/FULL SEARCH
/[NO]HEADER
/[NO]IO_PAGE
/LONG
/MAP[:filespec]
/[NO]MEMORY_MANAGEMENT[:n]
/OPTIONS[:optionsspec]
/OVERLAY DESCRIPTION
/POSTMORTEM
/[NO]PRINT
```

LINK (Cont.)

```

/PRIVILEGED[:n]
/[NO]RECEIVE
/[NO]RESIDENT_OVERLAYS
/SAVE
/[NO]SEGREGATE
/SEQUENTIAL
/SHAREABLE[:arg]
        COMMON
        LIBRARY
/SLAVE
/SLOW
/SYMBOL TABLE[:filespec]
/[NO]SYSTEM LIBRARY DISPLAY
/[NO]TASK[:filespec]
/TKB
/TRACE
/[NO]WARNINGS
/[NO]WIDE

```

Parameter

filespec[s]

Specifies the file or files to be linked. The default file type is .OBJ for object modules and .OLB for object module libraries.

You can have any number of object modules as input files, separated by commas. These can be specified in any order. If, however, one or more input files must reference an object library (.OLB) file, the .OLB file must be entered in the list of input files directly after the files that reference it.

Parameter Qualifiers

```

/[NO]CONCATENATE
/DEFAULT LIBRARY
/[NO]GLOBALS
/LIBRARY
/INCLUDE:(module1[:...:modulen])
/OVERLAY DESCRIPTION
/SELECTIVE_SEARCH

```

Command Qualifiers

/ANCILLARY_PROCESSOR[:n]

Identifies the task as an Ancillary Control Processor (ACP), which is a privileged task. The parameter n specifies the base relocation register. Acceptable values are 0, 4, or 5. The default is 5.

In TKB format, apply the /AC:n switch to the .TSK file specification.

LINK (Cont.)

```
/[NO]CHECKPOINT:arg
      SYSTEM
      TASK
```

Specifies that the task is to be (or is not to be) checkpointable. Checkpointability of tasks is an important part of the operating system's ability to share resources. When a higher priority task seeks access to system memory, a checkpointable task of lower priority is checkpointed, or rolled out to the disk to be stored in its current state until the higher priority task exits, whereupon the lower priority task returns and takes up where it left off.

If you do not use the /CHECKPOINT qualifier, your task is built noncheckpointable. The default is /NOCHECKPOINT. A noncheckpointable task cannot be dislodged by a task of higher priority. Therefore, you should always build your tasks checkpointable unless you have some important reason for not doing so.

You can specify how the checkpointing of the task is handled through the arguments to the /CHECKPOINT qualifier. Your task can be checkpointable to the system checkpoint file with the SYSTEM option. This is the default. LINK/CHECKPOINT and LINK/CHECKPOINT:SYSTEM are equivalent commands.

Tasks built with system checkpointing cannot be checkpointed if the system checkpoint file is full. (The size of this file is set with the SET DEVICE command.)

You can also reserve checkpoint space for the task as part of its own task image file by using LINK/CHECKPOINT:TASK. Such tasks are always checkpointable. (If there is no system checkpoint file, you can run only one copy of such tasks.)

It should be apparent that the checkpointability of a task has an impact on the efficient operation of the entire system as well as on the task itself. If the task is built noncheckpointable, it can block more important tasks from running. If it is built with task checkpointability, mass-storage space is reserved that may never be used. If it is built with system checkpointability, there may be no room for it in the system checkpoint file.

Check with your system manager for advice on the preferred selection among these checkpointing qualifiers.

In TKB format, for /CHECKPOINT:SYSTEM, apply the /CP switch to the task image file. For /CHECKPOINT:TASK, apply the /AL switch to the task image file. For /NOCHECKPOINT, apply the /-CP switch to the task image file.

```
/CODE:(arg[,s])
      EAE
      [NO]FPP
      PIC
      POSITION_INDEPENDENT
```

Specifies that the code for the task relies on certain hardware elements or employs certain techniques. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information.

LINK (Cont.)

The EAE argument specifies that the task uses the Extended Arithmetic Element. This hardware is optional and may not be part of your system. In TKB format, apply the /EA switch to the .TSK file specification.

The FPP argument specifies that the task uses the Floating Point Processor. This hardware is optional and may not be part of your system. The NOFPP argument is the default. In TKB format, apply the /FP switch to the .TSK file specification.

The PIC and POSITION_INDEPENDENT arguments are identical and specify that the resident common or library being built is position-independent. In TKB format, apply the /PI switch to the .TSK or .STB file specification.

If you want to use more than one argument, enclose them in parentheses, separated by commas.

/COMPATIBLE

Specifies that the task be built in compatibility mode. This means that memory-resident overlay segments are aligned on 256-word boundaries for compatibility with other implementations of the mapping directives. Without this qualifier, overlay segments are aligned on 32-word boundaries.

In TKB format, apply the /CM switch to the .TSK file specification.

/CROSS_REFERENCE

Specifies that a listing of symbol cross-references is to be appended to the Task Builder map file.

If you include this qualifier, LINK automatically includes the /MAP qualifier as well. You do not need to specify the /MAP qualifier unless you want to supply a name for the map file. If you supply a name, the map file appears in your current UFD but is not spooled to the line printer.

In TKB format, apply the /CR switch to the .MAP file specification.

/DEBUG[:filespec]

Specifies the inclusion of a debugging aid in the task image. A debugging aid permits you to interrupt the running of a task and inspect registers and other memory locations at various stages. If you give no file specification, the default is LB0:[1,1]ODT.OBJ, which is ODT, the On-Line Debugging Tool, a DIGITAL-supplied utility. ODT is incorporated in the task you are building. ODT can be used only on tasks written in the MACRO-11 Assembly Language. See the RSX-11M/M-PLUS and Micro/RSX Debugging Reference Manual for further information.

If you have a user-written debugger, name the file it is in when you use this qualifier. This file should be an object module.

LINK (Cont.)

In TKB format, apply the /DA switch to the .TSK file specification if you wish to use ODT. If you want to use a user-written debugger, apply the /DA switch to the input file specification naming the debugger.

/ERROR_LIMIT:n

Directs the Task Builder to abort LINK after n diagnostics errors have been produced.

In TKB format, apply the /XT:n switch to the .TSK file specification.

/[NO]EXECUTABLE[:taskname]

The /EXECUTABLE qualifier is a synonym for the /TASK qualifier. See the /TASK qualifier discussion for more information.

/FAST

Specifies that the Fast Task Builder be used to build the task. The Fast Task Builder is considerably faster than the default Task Builder.

Several restrictions apply to tasks to be built by the Fast Task Builder:

- Tasks may not be overlaid.
- Tasks may not be privileged.
- Tasks may not link to cluster libraries.
- Tasks may not link to resident libraries.
- Tasks may not use FORTRAN virtual arrays.
- The Task Builder cannot produce a .STB output file.
- The Task Builder cannot create resident libraries.
- The LINK command is limited to the following qualifiers:

/MAP, /WIDE

/[NO]CHECKPOINT:SYSTEM

/[NO]MEMORY_MANAGEMENT

/CODE:EAE

/CODE:[NO]FPP

/DEBUG[:filespec]

/LIBRARY

LINK (Cont.)

- The Task Builder is limited to the following options:

ACTFIL

ASG

COMMON

EXTSCT

LIBR

MAXBUF

RESCOM

RESLIB

STACK

TASK

UNITS

In addition, the map format is nonstandard and undocumented.

You can also invoke the Fast Task Builder from DCL with the command `RUN $FTB`. You must follow TKB format if you run the Fast Task Builder in this fashion.

See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more details.

/FULL_SEARCH

Specifies that when processing modules from the default object module library, the Task Builder should search all cotree (overlay) segments for matching definitions or references.

Without this qualifier, unintended global references between co-tree overlay segments are eliminated. Definitions of global symbols from the default library are restricted in scope to references in the main root and the current tree.

In TKB format, apply the `/FU` switch to the `.TSK` file specification.

/LONG

Specifies that the map file produced by the Task Builder include additional file information on modules used in the task build. The long map does not include file information on modules from the system library.

If you include this qualifier, LINK automatically includes the `/MAP` qualifier as well. You need not include the `/MAP` qualifier

LINK (Cont.)

unless you want to supply a name for the map file. If you supply a name, the map file appears in your current UFD but is not printed on the line printer.

In TKB format, apply the `/-SH` switch to the `.MAP` file specification.

`/MAP[:filespec]`

Specifies that you want a memory allocation, or map, file produced. If you use `/MAP` as a command qualifier, without a file specification `.argument`, the map appears in your UFD with a file name derived from the name of the first input file named in the command line and a file type of `.MAP`. The map is also spooled to the line printer.

If you use `/MAP` with a file specification argument, either as a command qualifier or a file specification qualifier, the map appears in your UFD with a file name you have given. The map is not spooled to the line printer.

If you use `/MAP` as a file specification qualifier, without a file specification argument, the map appears in your UFD with a file name derived from the name of the `.OBJ` or `.OLB` file to which the qualifier was attached. The map is not spooled to the line printer.

The following qualifiers automatically direct the Task Builder to produce a map with special characteristics:

```

/CROSS_REFERENCE
/LONG
/SYSTEM_LIBRARY_DISPLAY
/WIDE

```

You do not need the `/MAP` qualifier with these qualifiers unless you want to add a file specification argument to `/MAP`. These other map-related qualifiers can be used as either command or file specification qualifiers, but they have no effect on file names.

In TKB format, the `/MAP` qualifier corresponds to the second position in the list of TKB output files and has the default file type `.MAP`.

`/[NO]HEADER`

If you are building a shared region or a driver, you should specify this qualifier. The default is `/HEADER`.

In TKB format, apply the `/-HD` switch to either the `.TSK` or `.STB` file specification. See `/SHAREABLE`.

LINK (Cont.)

/[NO]IO_PAGE

/NOIO_PAGE indicates to the Task Builder that the task is over 12K and purposely does not map to the I/O page. This qualifier is for building privileged tasks only. The default is /IO_PAGE.

/[NO]MEMORY_MANAGEMENT[:n]

Specifies that the task is being built for a system with (or without) memory management hardware. Use the /NOMEMORY MANAGEMENT qualifier when you are building a task on a system with memory management to be run on an RSX-11M (or RSX-11S) system without the memory management hardware. n specifies the highest physical address of a task on the target system and can be 28 (the default) or 30.

The default is /MEMORY_MANAGEMENT.

In TKB format, apply the /-MM switch to the .TSK file specification.

/OPTION[:optionsspec]

Specifies one or more Task Builder options. Use this qualifier if you need to use any of the Task Builder options. Information on these options is summarized in Table 6-1. See Section 6.2.1 for information on using this qualifier. For full information, or options, see the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.

You can supply options for this qualifier in two ways: you can supply the file specification of a file containing the options, or you can specify the options themselves. If you do not include a file specification with this qualifier, you are prompted for options. If you want to enter multiple options, you must use a comma after each option listed. If you want to list multiple options on individual lines, you must end each line with a comma.

If you include a file specification with this qualifier, this file should contain only the option names, comments, and the arguments associated with the options, as shown:

```
UNITS=8
ASG=TT0:7:8
; COMMENTS, PRECEDED BY SEMICOLON, ALLOWED
PAR=KROBAR:50000:40000
```

This file should not include anything but the option statements and comments preceded by the semicolon (;). Comments preceded by the exclamation point (!) are not accepted in this instance. Do not put any slashes in this file.

Note that the ABORT option is the only means of preventing a task build once you have begun issuing commands to the Task Builder. A CTRL/Z simply directs the Task Builder to begin the task build based on whatever instructions you have issued up to that point. Thus, if you want to be sure of being able to stop a task build in this way, you must include the /OPTIONS qualifier in the LINK

LINK (Cont.)

command line. This assures the availability of the ABORT option. If you are typing the command line and make a mistake, specify the /OPTIONS qualifier, press RETURN, and enter the ABORT=0 option. Then press RETURN and the task build aborts.

/OVERLAY_DESCRIPTION

Specifies that the input file is an overlay description file (type .ODL) that controls the linking of the task. No other input file can be specified if you use this qualifier. The .ODL file specifies input files to LINK. /OVERLAY_DESCRIPTION can be either a command qualifier or a file specification qualifier.

In TKB format, use the .ODL file as the only input file to the right of the equal sign (=) with the /MP switch applied to the .ODL file specification.

/POSTMORTEM

Requests a Postmortem Dump should the task be terminated abnormally. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for information on Postmortem Dumps.

In TKB format, apply the /PM switch to the .TSK file specification.

/[NO]PRINT

Causes the map file to print on your printer. The /PRINT qualifier is the default operation. If you do not want a printed listing of the map file, use the /NOPRINT qualifier.

/PRIVILEGED[:n]

Indicates that task is privileged. The argument n specifies the base relocation register. Acceptable values are 0, 4, or 5. The default is 5.

In TKB format, apply the /PR switch to the .TSK file specification.

/[NO]RECEIVE

Indicates that the task may (or may not) receive messages by means of the Executive directive SEND. /RECEIVE is the default.

In TKB format, apply the /SE switch to the .TSK file specification.

/[NO]RESIDENT_OVERLAY

Enables (or disables) recognition of the memory-resident overlay operator (!) in an overlay description file. The qualifier is used with the /OVERLAY_DESCRIPTION qualifier when the task has memory-resident overlays. The default is /RESIDENT_OVERLAY.

In TKB format, apply the /RO switch to the .TSK file specification.

LINK (Cont.)

Table 6-1
Task Builder Options

Description and Language Key ¹	Format and Defaults ²
Control	
Direct TKB to terminate build (M,H)	ABORT=0 Number used for format only. Any number will do.
Identification	
Declare task name (M,H)	TASK=one through six alphanumeric characters TASK=first six characters of task image file name
Set task's User Identification Code (M,H)	UIC=[g,m] UIC=terminal UIC
Declare task's priority (M,H)	PRI=decimal integer 1-250 PRI=set by INSTALL; 50, for nonprivileged users
Declare task's partition name and size (M,H)	PAR=parname[:base:length] PAR=GEN
Declare number of open active files (H)	ACTFIL=decimal integer ACTFIL=4
Declare extension to FORTRAN record buffer (H)	MAXBUF=decimal integer for record buffer size in bytes (larger than default) MAXBUF=132.

1. Language key:

M - Option is of interest to MACRO-11 programmers.
H - Option is of interest to programmers using high-level languages.

2. The first entry in the format and default column gives the format, the second entry gives the default, if any. This chart is only a reminder of the options, their formats, and their defaults. You must read the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for full information on these options.

(continued on next page)

LINK (Cont.)

Table 6-1 (Cont.)
Task Builder Options

Description and Language Key	Format and Defaults
Identification (Cont.)	
Declare extension of buffer used for processing format strings (H)	FMTBUF=decimal integer for length of format buffer (larger than default) FMTBUF=132.
Declare extension of program section (M,H)	EXTSCT=psectname:extension No default
Declare extension of task's portion of memory; extends D-space of I- and D-space tasks (M,H)	EXTTSK=decimal integer for increase in task memory allocation (in words) Default from PAR option
Declare virtual base address and size of virtual program section (H)	VSECT=psectname:base>window[:length] No default
Declare number of additional window blocks needed by task (M,H)	WNDWS=1-7 WNDWS=0
Declare size of stack (M,H)	STACK=decimal integer for number of words needed for stack STACK=256.
Sharing Storage	
Declare intention to access system-owned common block for data in memory; if used with I- and D-space tasks, common mapped with D-space APRs only (M,H)	COMMON=name:access-code[:apr] No default
Declare intention to access system-owned resident library for code; if used with I- and D-space tasks, library mapped with I- and D-space APRs (M,H)	LIBR=name:access-code[:apr] No default

(continued on next page)

LINK (Cont.)

Table 6-1 (Cont.)
Task Builder Options

Description and Language Key	Format and Defaults
Sharing Storage (Cont.)	
Declare intention to access user-owned common; if used with I- and D-space tasks, common mapped with D-space APRs only (M,H)	RESCOM=filespec/access-code[:apr] Standard defaults on filespec
Declare intention to access user-owned library; if used with I- and D-space tasks, library mapped with I- and D-space APRs (M,H)	RESLIB=filespec/access-code[:apr] Standard defaults on filespec
Allow task to dynamically map memory-resident shared regions at run time. One task address window maps the libraries into the same span of virtual address space. (M,H)	CLSTR=library1,library2,...libraryn:switch:[apr] library1 is default library switch can be RW: or RO: apr is starting APR No default
Device	
Declare number of I/O (logical) units (M,H)	UNITS=decimal integer 0-250 UNITS=6
Declare logical device assignment to units (M,H)	ASG=ddnn:unit-num:unit-num . . . ASG=SY0:1:2:3:4,TI0:5,CL0:6
Alterations	
Declare definition of global symbol (M)	GBLDEF=symbol-name:symbol-value No default
Declare global symbol reference (M)	GBLREF=symbol-name No default
Declare series of patch values; patches I-space of I- and D-space tasks (M)	ABSPAT=seg-name:address:val-1 . . . (up to 8) No default

(continued on next page)

LINK (Cont.)

Table 6-1 (Cont.)
Task Builder Options

Description and Language Key	Format and Defaults
Alterations (Cont.)	
Declares series of patch values; patches D-space of I- and D-space tasks (M)	DSPPAT=seg-name:address:val-1 . . . (up to 8) No default
Declare series of patch values relative to global symbol (M)	GBLPAT=seg-name:sym-name[+/-offset]:val-1 . . . (up to 8) No default
Exclude the specified symbols from the symbol definition file of a supervisor mode library (M)	GBLXCL=sym1:sym2:...:symn No default
Declare address and size of ODT SST vector (M)	ODTV=symbol-name:vector-length No default
Declare size of task SST vector (M)	TSKV=symbol-name:vector-length No default
Include specified symbols in .STB file (M,H)	GLBINC=symbol-name,symbol-name,...,symbol-name No default

/SAVE

Specifies that DCL retains the indirect command file created by DCL to pass your LINK command to TKB. See Section 6.2.1 for a description of DCL's handling of the LINK command. If you include /SAVE in your LINK command line, a file named ATLNK.TMP appears in your directory after the task build completes. Because this file always has the same name, you should give it a name related to the task it builds, such as SHEMPBLD.CMD. Then you can issue a command in the following form:

```
$ LINK @SHEMPBLD (RET)
```

and duplicate the task build that originally produced the command file.

This file is also useful for comparing LINK command syntax with TKB syntax because it includes the full translation of the LINK command into TKB format.

LINK (Cont.)**/[NO]SEGREGATE**

Causes the Task Builder to order program sections alphabetically by name within access code (RO followed by RW). If you also specify /SEQUENTIAL, TKB orders program sections in their input order by access code.

/NOSEGREGATE is the default. TKB interleaves RO and RW program sections. When combined with /SEQUENTIAL, /NOSEGREGATE results in a task with program sections allocated in input order with its RW and RO sections interleaved. If you use /NOSEQUENTIAL and /NOSEGREGATE together, which is the default for both, TKB orders program sections alphabetically with RW and RO sections interleaved.

In TKB format, use the /[-]SG switch on the .TSK file specification.

/SEQUENTIAL

Directs Task Builder to construct a task image from program sections in the order in which they appear. Normally, the Task Builder finds all program sections referenced in all modules in an overlay segment and then builds the task with those program sections in alphabetical order. Do not use this qualifier to build tasks that rely on alphabetical allocation of program sections, such as FORTRAN I/O handling modules and FCS modules from SYSLIB. See also /SEGREGATE.

In TKB format, apply the /SQ switch to the .TSK file specification.

/SHAREABLE[:arg]

COMMON
LIBRARY

The COMMON argument informs the Task Builder that a shareable common is being built. You should always use the /NOHEADER qualifier with /SHAREABLE:COMMON. If you do not specify CODE:PIC or CODE:POSITION_INDEPENDENT, TKB builds an absolute shared common. All program sections in the common are marked absolute. If you specify CODE:PIC or CODE:POSITION_INDEPENDENT, all program sections in the common are marked relocatable. In either case, the .STB file contains all the program section names, attributes, lengths, and symbols. The .STB file of a common built /SHAREABLE contains all defined program sections. In TKB format, apply the /CO switch to the .TSK or .STB file specification.

The LIBRARY argument informs the Task Builder that a shareable library is being built. You should always use /NOHEADER with /SHAREABLE:LIBRARY. TKB includes only one program section in the .STB file. If you do not specify CODE:PIC or CODE:POSITION_INDEPENDENT, TKB names the program section .ABS, makes the library position-dependent, and defines all symbols as absolute. If you specify CODE:PIC or CODE:POSITION_INDEPENDENT, TKB gives the program section the same name as the root segment of the library. TKB forces this name to be the first and only

LINK (Cont.)

declared program section in the library. TKB declares all global symbols in the .STB file relative to that program section. In TKB format, apply the /LI switch to the .TSK or .STB file specification.

/SLAVE

Specifies that the task is to be slaved to a sending task. When a slaved task successfully executes the Executive directive Receive Data, it is given the UIC and TI: of the sending task. This qualifier applies only to systems with multiuser protection.

Slaved tasks cannot be run with a RUN command. They must be run by the sending task.

In TKB format, apply the /SL switch to the .TSK file specification.

/SLOW

Specifies that you want to use the slower, but more economical method of task building. You should specify this qualifier only if your link or task build produces the following error message:

NO VIRTUAL MEMORY STORAGE AVAILABLE

The Task Builder (without the /SLOW qualifier) and the Fast Task Builder use a symbol table structure that can be searched quickly, but which requires more work file space than if you specified the /SLOW qualifier. If you receive the error message shown, you can either reduce the work-file size (as described in the Task Builder manual) or use the /SLOW qualifier.

In TKB format, apply the /SB switch to the .TSK file specification.

/SYMBOL_TABLE[:symboltablespec]

Directs that a symbol table file be produced. The default name is that of the first input file and the default type is .STB. The file specification parameter overrides the defaults. This qualifier is used when building shared regions.

In TKB format, this qualifier corresponds to the third position in the list of TKB output files, called the .STB file specification.

/[NO]SYSTEM_LIBRARY_DISPLAY

Directs the Task Builder to produce a map that includes (or does not include) global symbols found in LB0:[1,1]SYSLIB or in any shared regions to which the task links. This map is usually considerably longer than the default map. The information displayed illuminates the contribution that SYSLIB or the shared regions make to the task.

If you include this qualifier, LINK includes the /MAP qualifier as well. You do not need the /MAP qualifier unless you want to supply a name for the map file. If you supply a name, the map file appears in your current UFD but is not spooled to the line printer.

LINK (Cont.)

See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information.

In TKB format, apply the /MA switch to the .MAP file specification.

/[NO]TASK[:taskspec]

Specifies a name for the task image file different from that of the first input file plus the type .TSK. If used as a file specification qualifier, the task name is derived from the name of the file to which the qualifier is attached. /EXECUTABLE is a synonym for /TASK.

In TKB format, this qualifier corresponds to the first position in the list of TKB output files, called the .TSK file specification.

The /NOTASK qualifier specifies that LINK produce no task image file. This qualifier is useful when you want to use some facility of the Task Builder without building a task, to check for unresolved symbol references or make a map, for instance. /NOEXECUTABLE is a synonym.

In TKB format, leave the first position in the list of TKB output file specifications blank, followed by a comma (,).

/TKB

Specifies that the default Task Builder be used to build the task. This is the default; the qualifier is included for completeness.

You can also invoke the default Task Builder from DCL with the command RUN \$TKB. You must follow TKB format if you run the default Task Builder in this fashion.

/TRACE

Specifies that the task is to be traceable. When you use this qualifier, a trace trap occurs on the completion of each instruction when the task is run.

In TKB format, apply the /TR switch to the .TSK file specification.

/[NO]WARNINGS

Suppresses diagnostic messages issued by the the Task Builder. Two messages are suppressed:

n undefined symbols segment "segname"

and

Module "modulename" multiply defines P-section "psectname"

The default is /WARNINGS.

In TKB format, use the /[-]NM switch on the .TSK file specification.

LINK (Cont.)

/[NO]WIDE

Specifies that the Task Builder map be printed in 132-column format. The default is /WIDE.

If you include this qualifier, you include the /MAP qualifier as well. You do not need the /MAP qualifier unless you wish to supply a name for the map file. If you supply a name, the map file appears in your current UFD but is not spooled to the line printer.

In TKB format, apply the /WI switch to the .MAP file specification.

Parameter Qualifiers

Any input file can have a parameter qualifier applied to it, identifying the kind of file that it is.

/[NO]CONCATENATE

Identifies the input file as a concatenated object file; this is the default. All modules in the file are processed to form the task image. /NOCONCATENATE specifies that only the first object module encountered is to be processed, regardless of how many are present.

In TKB format, apply the /CC switch to an input file containing concatenated object modules.

/DEFAULT_LIBRARY

Specifies that the file to which it is appended replace the system object module library, LB0:[1,1]SYSLIB.OLB, as the default library that is searched to resolve unresolved global references. This qualifier can be applied to only one file and that file must be an object module library, type .OLB.

In TKB format, apply the /DL switch to an input library file.

/[NO]GLOBALS

Specifies that global symbols referenced and defined by the input file are (or are not) to be included in the map output file. The default is /GLOBALS.

In TKB format, apply the /-MA switch to the input file.

/INCLUDE:module1[:...modulen]

Specifies as many as eight module names from a library using /INCLUDE. You must specify at least one. If you use the optional module arguments, the Task Builder takes only those modules from the library. The module names are defined at assembly time.

If you want both to resolve undefined references to global symbols and to specify modules, you must use this qualifier twice.

LINK (Cont.)

In TKB format, apply the /LB switch to an input library file in two different ways for both /LIBRARY and /INCLUDE.

/LIBRARY

Identifies the file as an object module library. This qualifier is required for any input library file and is prohibited for any other type of file. The default file type for object libraries is .OLB.

The Task Builder searches the library file to resolve all undefined global symbol references from files appearing to the left of the library file in the LINK command line. The Task Builder then extracts any and all modules that resolve undefined references and includes them in the task image. See also /INCLUDE.

/SELECTIVE_SEARCH

Instructs the Task Builder to search the file only for undefined references to global symbols. This qualifier is most useful when building an Ancillary Control Processor (ACP) or other privileged task that maps into the Executive. If you do not specify this qualifier, all the Executive's global symbol definitions are included in the task build, whether there are undefined references to the global symbol or not. The Executive contains a myriad of modules. In these and similar circumstances, this qualifier considerably shortens the symbol table search and improves system performance.

If you do not use this qualifier, all global symbols from the input file are included in the task image.

In TKB format, apply the /-SS switch to an input file.

Examples

```
$ LINK (RET)
File(s)? WRAY (RET)
$
```

This example task builds the object file WRAY.OBJ into the task image file WRAY.TSK. For the LINK command to work properly, WRAY.OBJ must be an object file. Because the user did not specify a file type for the object file, the LINK command supplied the default file type .OBJ.

The return of the implicit prompt indicates successful completion of the task build.

```
$ LINK WRAY (RET)
```

This example is equivalent to the previous one.

LINK (Cont.)

```
$ LINK/MAP WRAY (RET)
```

This example is similar to the previous two except that a Task Builder map is also requested. When the task build is completed, the map WRAY.MAP is spooled to the line printer and also appears in your directory.

```
$ LINK/MAP:NEWLINK WRAY (RET)
```

This example is similar to the previous one, except that a file specification is provided with the /MAP qualifier. When the task build is completed, the map file NEWLINK.MAP appears in your directory. If you want a copy of this file, issue a PRINT command.

```
$ LINK WRAY/MAP (RET)
```

This example illustrates another way of using the /MAP qualifier. In this case, when the Task Builder exits, a file called WRAP.MAP appears in your directory but is not spooled to the line printer. If you want a copy of this file, issue a PRINT command.

```
$ LINK/DEBUG WRAY (RET)
```

This example builds the task WRAY.TSK including ODT, the On-line Debugging Tool.

```
$ LINK/OPTIONS WRAY (RET)
```

```
Option?UNITS=8 (RET)
```

```
Option?ASG=TT0:7:8 (RET)
```

```
Option? (RET)
```

```
$
```

This example illustrates the use of the /OPTIONS qualifier. The command sequence builds the task with eight logical units (LUNs), with LUNs 7 and 8 assigned to TT0:. See the discussion in Section 6.2.1. The entire LINK command is entered by the final RETURN in response to the Option? prompt.

```
$ LINK/OPTIONS/NOMEMORY_MAN WRAY (RET)
```

```
Option?PAR=KROBAR:50000:40000 (RET)
```

```
Option? (RET)
```

```
$
```

This example builds the task to run on an unmapped system in the partition named KROBAR, which has a base address of 50000 and a size of 40000.

```
$ LINK/CHECKPOINT:SYSTEM WRAY (RET)
```

```
$ LINK/CHECKPOINT WRAY (RET)
```

The commands in this example are equivalent. The task is built checkpointable to the system checkpoint file.

LINK (Cont.)

```
$ LINK/CHECKPOINT:TASK WRAY (RET)
```

In this example, the task is built with checkpoint space reserved where the task image file is stored.

```
$ LINK WRAY, 3TRACK (RET)
```

In this example, the task is built from two object modules, WRAY.OBJ and 3TRACK.OBJ. The task image file has the name WRAY.TSK. It is installed and run under that name.

```
$ LINK/TASK:SHACK WRAY, 3TRACK (RET)
```

In this example the task is built from the same two object modules as in the previous example. However, the task image file is given the name SHACK.TSK by the /TASK qualifier. It is installed and run under that name.

```
$ LINK/OPTIONS WRAY, 3TRACK (RET)
Option?TASK=SHACK (RET)
Option? (RET)
$
```

In this example, the task is built from two object modules. The task image file has the name WRAY.TSK, but the default task name (specified by the TASK option) is SHACK. The default task name is used by the INSTALL command (see Section 7.8) if the user does not specify a task name explicitly. Only privileged users can issue the INSTALL command. This means that to run the task from a file, you use the command RUN WRAN, but if you run it after it has been installed, you use the command RUN SHACK.

```
$ LINK/OVERLAY RAYMEN (RET)
```

In this example, the Task Builder uses the file RAYMEN.ODL to build an overlaid task. (.ODL is the default file type for the /OVERLAY qualifier.) The input file RAYMEN.ODL must be written in the Overlay Description Language. Only one file can be specified with this qualifier. The files that form the overlaid task are called automatically.

```
$ LINK @WRAYBLD (RET)
```

This example specifies the indirect command file WRAYBLD.CMD as input to the Task Builder. This file must be written in Task Builder format, not DCL format. This file can be the ATLNK.TMP file renamed. See the discussion under /SAVED.

Notes

You must read the appropriate documentation to use this command fully. This is true without exception.

The RSX-11M/M-PLUS and Micro/RSX Task Builder Manual does not use the term "link." In the Task Builder manual, the preferred term is "build."

LINK (Cont.)

Note that nonprivileged users can build privileged tasks and high-priority tasks, but those users cannot run the privileged tasks. Nor can they run tasks at any but the default priority of 50.

Remember that for nonprivileged users, the RUN command is always a combination of three commands. In effect, it is an INSTALL-RUN-REMOVE command. Thus, references to installation in the preceding discussion refer to explicit installation by privileged users through the INSTALL command and implicit installation through the RUN command as well. See Chapter 7 for further details.

You can include comments in the LINK command line in the usual DCL fashion. However, comments in a list of TKB options must appear on a line by themselves preceded by a semicolon. Do not use the exclamation point (!) for comments in a list of options.

Error Message

LIN -- Library invalid on last input file

Explanation: The syntax of LINK requires that the last input file specified must be a valid (single) object file that is not part of a library.

User Action: Check for proper syntax and reenter the command.

LIBRARY

6.3 LIBRARY

LIBRARY creates and maintains user-written library files, or libraries. Libraries can contain macro definitions, object modules, or, in the case of universal libraries, anything.

Maintenance functions of LIBRARY include extracting, deleting, inserting, and replacing library entries. Libraries can be compressed (squeezed) to save disk space. Library contents can also be listed.

User-written object module libraries can be used as input to the Task Builder, either directly or through LINK. The Task Builder (TKB) searches for definitions of all global symbols referenced in your program in the following fashion:

1. Searches other modules included in the LINK command
2. Searches a user-written object module library included in the LINK command
3. Searches the system library

User-written macro libraries hold source macros for use as input by the MACRO-11 Assembler, either directly or through MACRO. The assembler searches the library you specify for macros listed in .MCALL statements and called in your source program before searching the system macro library.

In general, LIBRARY handles object module libraries and macro libraries in the same way. This means that references in this section to modules can be usually taken as referring to macros as well as object modules.

Libraries are direct access files containing one or more modules of the same type. These files are organized for rapid access by the Task Builder and MACRO-11 Assembler. Universal libraries can contain virtually anything to which you wish to have rapid access, such as text files.

Library files contain two directory tables. The entry point table (EPT) contains entry point names (global symbols). The module name table (MNT) contains module names. Both tables are in alphabetical order. Object module names are derived from .TITLE directives, while entry point names are derived from defined global symbols. Once an entry point is located, its associated module can be accessed directly. Macro module names are derived from .MACRO directives; macros are linear code and do not have entry points.

This section does not teach you how to use the Librarian Utility Program (LBR), but only how to use the LIBRARY command. See the Notes for suggested further reading.

Format

```
LIBRARY
Operation? operation[/qualifier[s]]

LIBRARY operation[/qualifier[s]]

LIBRARY @filespec
```

LIBRARY (Cont.)

LIBRARY performs the following operations:

COMPRESS	CREATE	DELETE	EXTRACT
INSERT	LIST	REMOVE	REPLACE

Each of these operations is treated as a separate command in this description.

Notes

LIBRARY invokes the Librarian Utility Program (LBR). To use this command successfully, you must familiarize yourself with this utility, which is described in the RSX-11M/M-PLUS Utilities Manual.

LBR format differs from the format of the DCL LIBRARY command. DCL translates the DCL command line into LBR format. The descriptions presented here include cross-references to the equivalents in LBR format of the DCL command elements. Use the SET DEBUG command to see the translation.

If you wish to use indirect command files with LIBRARY, the indirect command file must be in LBR format, not DCL format. Only one level of indirect referencing is permitted.

You should also be familiar with the way the MACRO-11 Assembler and the Task Builder use libraries. See the PDP-11 MACRO-11 Language Reference Manual, the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual, and the RSX-11M/M-PLUS Guide to Program Development for more information.

You cannot use wildcards (the asterisk(*) and the percent sign (%)) in file specifications in LIBRARY operations.

Library modules are limited to 65536 words per module.

Libraries are limited to 65536 blocks.

The LIBRARY command operates on three kinds of libraries:

- Object module libraries. These can be user libraries or one of the following DIGITAL-supplied libraries:

LB0:[1,1]SYSLIB.OLB	System library (this is the library searched by the Task Builder by default).
LB0:[1,1]VMLIB.OLB	Virtual memory management library.
LB0:[1,1]EXELIB.OLB	Executive library.
LB0:[1,1]RMSLIB.OLB	Record Management Services library.
LB0:[1,1]FOROTS.OLB	FORTTRAN IV library (optional).
LB0:[1,1]F4POTS.OLB	FORTTRAN-IV-PLUS and FORTTRAN-77 library (optional).

LIBRARY (Cont.)

- Macro libraries. These can be user libraries or one of the following DIGITAL-supplied libraries:

LB0: [1,1]RSXMAC.SML System macro library. This is the library searched by the assembler by default (note nonstandard file type).

LB0: [1,1]EXEMAC.MLB Executive macro library.

LB0: [1,1]RMSMAC.MLB Record Management Services library.

- Universal libraries. The Librarian utility can also be used to create and maintain a third form, universal libraries, using the /UNIVERSAL qualifier on some of the LIBRARY commands.

The LIBRARY command and utility have no relationship to the other kinds of libraries used on RSX-11M systems. Specifically, LBR has no relationship to supervisor-mode libraries or core-resident libraries created by the Task Builder.

On RSX-11M, the system directory (LB0: [1,54], which contains all system tasks, such as commands and utilities) is also called a library, but also has no relationship to the LIBRARY command.

The /SQUEEZE qualifier causes all text to the right of the rightmost semicolon to be deleted. This convention permits you to preserve any meaningful semicolon by adding another semicolon to its right. Examples of meaningful semicolons include using the ASCII value of the semicolon in code, or a semicolon marking a comment you wish to preserve. The following line of code moves the ASCII value of the semicolon to register 1:

```
MOV B #' ;, R1
```

but emerges after the /SQUEEZE as follows:

```
MOV B #'
```

To preserve the meaningful semicolon, use the following form:

```
MOV B #' ;, R1 ; THIS DUMMY COMMENT SAVES THE CODE
```

which emerges after the /SQUEEZE as follows:

```
MOV B #' ;, R1
```

Similarly, the following comment:

```
;NEXT 5 LINES ARE CRUCIAL
```

disappears after the /SQUEEZE, but this form:

```
;NEXT 5 LINES ARE CRUCIAL ;
```

emerges after the /SQUEEZE as the following:

```
;NEXT 5 LINES ARE CRUCIAL
```

LIBRARY (Cont.)

Keep this convention in mind whenever you are using the /SQUEEZE qualifier.

All error messages are described in Section 6.3.9. Some error messages are closely related to particular LIBRARY operations. Descriptions of these messages are repeated in the section on the operation to which they pertain.

6.3.1 LIBRARY/COMPRESS

LIBRARY/COMPRESS physically deletes modules that have been logically deleted through LIBRARY/DELETE. You can rename the resulting compressed library.

Format

LIBRARY/COMPRESS[:(arg[,s])] libspec [newlibspec]

Arguments

GLOBAL:n
MODULES:n
BLOCKS:n

Parameters

libspec

Specifies the name of the library file to be compressed. The default type is .OLB, specifying an object module library. If you wish to compress a macro library (standard file type .MLB), or a universal library (standard file type .ULB), you must specify the file type explicitly.

newlibspec

This optional parameter specifies a name for the newly compressed library. If you do not specify a name, the new file has the same name as the old one. The old file is not deleted after you create a new library with this command.

Arguments

If you are specifying more than one argument, the arguments must be enclosed in parentheses and separated by commas. If you are specifying only one argument, the parentheses are not necessary.

GLOBALS:n

Specifies the number of global symbols (entry point table entries) to allocate. The default n is the number of global symbols allocated in the old library. The maximum n is 4096. The value of n is always forced to zero for macro and universal libraries.

LIBRARY (Cont.)

MODULES:n

Specifies the number of entries to allocate in the module name table. The default value is the number of entries in the old library. The maximum number of module names is 4096.

BLOCKS:n

Specifies the size of the library in 256-word blocks. The default size is the size of the old library.

Example

```
$ LIBRARY/COMPRESS LB:[1,1]SYSLIB.OLB (RET)
```

This example compresses the system object module library.

Notes

This command is equivalent to the LBR /CO switch applied to a library file.

If you like, you can use LIBRARY/LIST before and after compressing a library to see the effects of the compress operation.

6.3.2 LIBRARY/CREATE

LIBRARY/CREATE creates a library and optionally inserts one or more modules into it.

The library is a contiguous file created on a volume mounted on a random-access device, that is, disk or DECTape.

Format

```
LIBRARY/CREATE[:(arg[,s])][qualifier[s]] libspec [infilespec[,s]]
```

Arguments

GLOBALS:n

MODULES:n

BLOCKS:n

Command Qualifiers

/[NO]GLOBALS

/MACRO

/OBJECT

/SELECTIVE_SEARCH

/SQUEEZE

/UNIVERSAL

LIBRARY (Cont.)**Parameters****libspec**

Specifies the name of the library file being created. The default file type is .OLB if an object module library is being created, .MLB if a macro library is being created, and .ULB if a universal library is being created.

infilespec[s]

Specifies the file or files to be used as input to the new library file. If no input files are specified, an empty library file is created. The default file types are .OBJ when creating object module libraries, .MAC when creating macro libraries, and .UNI when creating universal libraries.

Arguments

If you are specifying more than one argument, the arguments must be enclosed in parentheses and separated by commas. If you are specifying only a single argument, the parentheses are not necessary.

GLOBALS:n

Specifies the number of global symbols (entry point table entries) to allocate. The default is 512 for object libraries. This value is always forced to 0 for macro and universal libraries. n can be from 0 through 4096.

MODULES:n

Specifies the number of entries to allocate in the module name table. The default value is 256. n can be from 0 through 4096.

BLOCKS:n

Specifies the size of the library in 256-word blocks. The default size is 100 blocks.

Command Qualifiers**/[NO]GLOBALS**

Specifies whether global symbols should be included in the entry point table.

Use this qualifier if you wish to use the same global symbols in more than one module. /GLOBALS is the default but is a no-op.

/SELECTIVE_SEARCH

Sets the selective search attribute bit in the module header of object modules as they are inserted into an object library. You must specify an input file or files.

This qualifier has meaning for object module libraries only.

LIBRARY (Cont.)

Object modules with the selective search attribute are given special treatment by the Task Builder. Global symbols defined in modules with the selective search attribute are only included in the Task Builder's symbol table if they were previously referenced by other modules.

/SQUEEZE

Reduces the size of macro definitions by eliminating all trailing blanks and tabs, blank lines, and comments from macro text. You must specify an input file or files.

This qualifier has meaning for macro libraries only.

Macros that have been squeezed not only take up less room in the macro library file, but also take up less memory.

See the Notes to Section 6.3 for important information on the /SQUEEZE qualifier.

This /SQUEEZE qualifier is the equivalent of the LBR /SZ switch applied to the output library file when using the /CR switch.

/MACRO

Specifies that the library being created is a macro library.

/OBJECT

Specifies that the library being created is an object module library. This is the default and need not be specified.

/UNIVERSAL

Specifies that the library being created is a universal library.

Example

```
$ LIBRARY (RET)
Operation? CREATE:(GLOBALS:128,MODULES:64) (RET)
Library? CARNEGIE (RET)
Modules? DEWEY,CONGRESS,MODERN (RET)
```

This example creates the library CARNEGIE.OLB with a size of 100 blocks (the default), 128 entry points, and 64 module names, and inserts object modules from the input files DEWEY.OBJ, CONGRESS.OBJ, and MODERN.OBJ.

Notes

This command is the equivalent of applying the LBR /CR switch on the output file.

Tables should be allocated to maximum anticipated size. Expanding table allocations requires using LIBRARY/COMPRESS to copy the entire file.

If the qualifiers /SELECTIVE_SEARCH and /SQUEEZE are used with LIBRARY/CREATE, you must specify an input file or files.

LIBRARY (Cont.)**6.3.3 LIBRARY/DELETE**

LIBRARY/DELETE deletes modules from any library. See LIBRARY/REMOVE for removing global symbols (entry points) from a library.

Format

```
LIBRARY/DELETE libspec module[,s]
```

Parameter**libspec**

Specifies the name of the library from which modules are to be deleted.

module[,s]

You can specify as many as 15 names of modules to be deleted, separated by commas.

If you do not specify the module list, you will be prompted for it.

Example

```
$ LIBRARY/DELETE (RET)
Library? BUMPERS (RET)
Module(s)? BUGGY, BABY, RUBBER (RET)
```

Modules deleted:

```
BUGGY
BABY
RUBBER
```

This example deletes the modules BUGGY, BABY, and RUBBER from the latest version of the object module library BUMPERS.OLB.

Notes

This command is the equivalent of the LBR /DE switch applied to the output file.

When you delete an entry from a library, the entry is not physically removed, but is marked for deletion. This means that although the module is no longer accessible, the file space it occupied is not available. To clear this space, use the LIBRARY/COMPRESS operation, Section 6.3.1.

If a specified module is not contained in the library, the following message is printed on your terminal:

```
LBR -- *FATAL*--No module named "name"
```

All the modules in your list up to the incorrect module name will have been deleted. None of the modules following the incorrect module name will have been deleted. You should use LIBRARY/LIST, Section 6.3.6, to check which modules have been deleted and which have not.

LIBRARY (Cont.)

See LIBRARY/REMOVE (Section 6.3.7) for information on removing global symbols from libraries.

6.3.4 LIBRARY/EXTRACT

LIBRARY/EXTRACT reads one or more modules from a library and writes them into a specified output file.

You can extract up to eight modules with a single LIBRARY/EXTRACT operation. If more than one module is extracted, the modules are concatenated in the output file.

Format

```
LIBRARY/EXTRACT/qualifier[:filespec] libspec module[,s]
```

Command Qualifier

```
/OUTPUT[:filespec]
```

Parameters

libspec

Specifies the library from which modules are to be extracted.

The default file type is .OLB, corresponding to an object module library. If you wish to extract from a macro or universal library, you must specify the appropriate file type.

module[,s]

Specifies the modules that are to be extracted. If you do not include a list, all modules in the library are extracted and concatenated in the output file in alphabetical order. You can specify up to eight modules, separated by commas.

Command Qualifier

/OUTPUT

Specifies the file to which the extracted modules or macros are to be written. If you specify /OUTPUT without a file specification, the default is to write the modules to your terminal. This makes sense only for macro libraries or universal libraries containing text modules.

If you do not include the qualifier, you will be prompted To?, to which you are to reply with a file specification. You can reply TI: to have the output printed on your terminal.

Examples

```
$ LIBR/EXTRACT/OUTPUT:MATH LB0:[1,1]SYSLIB ARITH (RET)
```

This example extracts the module named ARITH from the library SYSLIB.OLB, the system object library. The module is placed in the default directory as a file named MATH.OBJ.

LIBRARY (Cont.)

```

$ LIBR/EXTRACT/OUT:TI: LB0:[1,1]RSXMAC.SML STOP$$ (RET)
  .MACRO  STOP$$  ERR
  .MCALL  DIR$
  MOV     (PC)+,-(SP)
  .BYTE  131.,1
  DIR$   ,ERR
  .ENDM   STOP$$

```

This example extracts the macro named STOP\$\$ from the system macro library and prints it on the terminal.

Notes

This command is the equivalent of the LBR /EX switch applied to the library file as an input file.

LIBRARY/EXTRACT has no effect on the library from which the modules are extracted.

6.3.5 LIBRARY/INSERT

LIBRARY/INSERT inserts modules from one or more files into a library.

Format

```
LIBRARY/INSERT libspec filespec[,s]
```

Command Qualifiers

```

/[NO]GLOBALS
/SELECTIVE_SEARCH
/SQUEEZE

```

Parameters**libspec**

Specifies the library into which modules are to be inserted.

The default file type is .OLB. If you wish to insert modules into a macro library (file type .MLB) or a universal library (file type .ULB), you must specify the file type explicitly.

filespec[s]

Specifies the file or files containing concatenated modules to be inserted into the specified library.

Any number of files can be specified and each file can contain any number of concatenated modules.

The default file type is determined by the library into which modules are being inserted. It is .MAC for macro libraries, .OBJ for object module libraries, and .UNI for universal libraries.

LIBRARY (Cont.)

Command Qualifiers

/[NO]GLOBALS

Specifies whether entry points for the specified modules should be included in the entry point table.

Use this qualifier if you wish to insert global symbols having the same name as symbols already in the library file. The default, /GLOBALS, does not permit this operation.

/SELECTIVE_SEARCH

Sets the selective search attribute bit in the module header of object modules as they are inserted into an object library. You must specify an input file or files.

Object modules with this attribute are given special treatment by the Task Builder. Global symbols defined in modules with the selective search attribute are not included in the Task Builder's symbol table unless they were previously referenced by other modules.

/SQUEEZE

Reduces the size of macro definitions by eliminating all trailing blanks and tabs, blank lines, and comments from macro text. You must specify an input file or files.

Macros that have been squeezed not only take up less room in the macro library file but also take up less memory.

See the Notes to Section 6.3 for important information on the /SQUEEZE qualifier.

This qualifier is the equivalent of the LBR /SZ switch applied to the output file.

Example

```
$ LIBR/INSERT (RET)
Library? LB0:[1,1]SYSLIB (RET)
Module(s)? USERSUB (RET)
```

This example inserts object modules from the file USERSUB.OBJ into the system object module library, LB0:[1,1]SYSLB.OLB.

Notes

This command is the equivalent of the LBR /IN command.

Note that a single input file can contain more than one object module or macro.

If you are inserting macros into a macro library, LIBRARY/INSERT insert only the .MACRO and .ENDM lines and what appears between them. All text in the file not bracketed by these assembler directives is ignored. This feature is in addition to the action of the /SQUEEZE qualifier.

LIBRARY (Cont.)

If you attempt to insert a module that already exists in the library file, the following message is printed on your terminal:

```
LBR -- *FATAL*Duplicate module name "name" in filename
```

If you attempt to insert a module with an entry point that duplicates one that is already in the EPT, the following message is printed on your terminal:

```
LBR -- *FATAL*Duplicate entry point "name" in filename
```

In the case of either of these messages, all operations up to the point at which the error occurred should have been successful. No operations past that point will have taken place. Use LIBRARY/LIST to check which modules have been inserted and which have not.

6.3.6 LIBRARY/LIST

LIBRARY/LIST lists the names of all modules in a library on your terminal or in an output file.

Format

```
LIBRARY/LIST[:filespec] libspec
```

Command Qualifiers

```
/BRIEF
/FULL
/[NO]NAMES
```

Parameter

libspec

Specifies the library you wish listed. The default file type is .OLB. If you wish to list a universal or macro library, you must specify the appropriate file type.

Command Qualifiers

```
/LIST[:filespec]
```

If you do not include a file specification as an argument to /LIST, the library is listed on your terminal.

If you include a file specification as an argument, the default device and UFD are the same as those of the library you are listing and NOT your current defaults. See the examples.

```
/BRIEF
```

Specifies that you wish the list to include only the module names. This is the default. See the examples.

LIBRARY (Cont.)**/FULL**

Requests a listing of all module names, along with a module description, including size, date of insertion, and module-dependent information. See the examples.

/[NO]NAMES

Requests a list of modules in the library, along with their entry points. The default is the /NONAMES qualifier. See the examples.

Examples

```
$ LIBRARY/LIST LBO:[1,1]SYSLIB (RET)
```

```
Directory of file SYSLIB.OLB;1002
Object module library created by: LBR V06.00
Last insert occurred 25-MAY-85 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words
```

```
ALERR
ALSCT
ALTPRI
ANSPAD
ARITH
.
.
.
```

This example shows the default display from LIBRARY/LIST. Only the module names appear. The information appears on your terminal.

```
$ LIBRARY/LIST/FULL LBO:[1,1]SYSLIB (RET)
```

```
Directory of file SYSLIB.OLB;1002
Object module library created by: LBR V06.00
Last insert occurred 25-MAY-85 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words

ALERR   Size:00073   Inserted:22-MAY-85   Ident:00
ALSCT   Size:00134   Inserted:22-MAY-85   Ident:00
ALTPRI  Size:00081   Inserted:22-MAY-85   Ident:00
ANSPAD  Size:00068   Inserted:25-MAY-85   Ident:20.0P6
ARITH   Size:00088   Inserted:22-MAY-85   Ident:03.03
.
.
.
```

LIBRARY (Cont.)

This example shows the display from LIBRARY/LIST/FULL, which includes full module descriptions. This information appears on your terminal.

```
$ LIBRARY/LIST/NAMES LB0:[1,1]SYSLIB (RET)
```

```
Directory of file SYSLIB.OLB;1002
Object module library created by: LBR V06.00
Last insert occurred 25-MAY-85 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words
```

```
** Module:ALERR
```

```
  $ALERR
```

```
** Module:ALSCT
```

```
  ALSCT
```

```
** Module:ALTPRI
```

```
  ALTPRI
```

```
** Module:ANSPAD
```

```
  ..ANSP
```

```
** Module:ARITH
```

```
  $DIV  $MUL
```

```
  .
  .
  .
```

This example shows the display from LIBRARY/LIST/NAMES, which lists entry points for each module.

```
$ LIBRARY/LIST/FULL/NAMES LB0:[1,1]SYSLIB (RET)
```

```
Directory of file SYSLIB.OLB;1002
Object module library created by: LBR V06.00
Last insert occurred 25-MAY-85 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words
```

LIBRARY (Cont.)

** Module:ALERR Size:00073 Inserted:22-MAY-85 Ident:00

\$ALERR

** Module:ALSCT Size:00134 Inserted:22-MAY-85 Ident:00

ALSCT

** Module:ALTPRI Size:00081 Inserted:22-MAY-85 Ident:00

ALTPRI

** Module:ANSPAD Size:00068 Inserted:25-MAY-85 Ident:20.OP6

..ANSP

** Module:ARITH Size:00088 Inserted:22-MAY-85 Ident:03.03

.
.
.

This example shows the display from LIBRARY/LIST/FULL/NAMES, which includes not only module descriptions but also entry points for each module.

```
$ SHOW DEFAULT (RET)
DB2: [303,5] TT56:
$ LIBRARY/LIST:FRED.LIS LB0:[1,1]SYSLIB (RET)
LBR -- *FATAL*-Open failure on file FRED.LIS
$
```

In this example, the user intended to create a list of the modules in SYSLIB in the default UFD on the default device. The error message indicates that the listing file cannot be opened. This is because LBR attempted to place FRED.LIS on LB0: in [1,1], where the user did not have file creation privileges. See next example.

```
$ SHOW DEFAULT (RET)
DB2: [303,5] TT56:
$ LIBRARY/LIST:DB2:[303,5]FRED.LIS LB0:[1,1]SYSLIB (RET)
$
```

In this example, the user explicitly stated the device and directory for the listing file and the command succeeded.

LIBRARY (Cont.)

Notes

This command is the equivalent of applying the LBR /LI, /FU, or /LE switches to the listing file.

Note that there are four possible listing formats with this command.

6.3.7 LIBRARY/REMOVE

LIBRARY/REMOVE removes global symbols (entry points) from a library. See LIBRARY/DELETE for deleting object modules from a library.

Format

```
LIBRARY/REMOVE libspec global [global[,s]]
```

Parameters

libspec

Specifies the name of the library from which you wish to remove global symbols.

global[,s]

Specifies the global symbols you wish to remove from the library. You can specify as many as 15 global symbols.

Example

```
$ LIBRARY/REMOVE DOUBLE TINKER,EVERS,CHANCE (RET)
```

Entry points deleted:

```
TINKER
EVERS
CHANCE
```

This example deletes the globals TINKER, EVERS, and CHANCE from the library DOUBLE.OLB.

Notes

This command is the equivalent of the LBR /DG switch applied to the output file.

If a specified global is not included in the library, the following message is printed on your terminal:

```
LBR -- *FATAL* -- No entry point named "name"
```

All the global symbols (entry points) in your list up to the incorrect name will have been deleted. None of the globals following the incorrect name will have been deleted. You should use LIBRARY/LIST (Section 6.3.6) to check which globals have been deleted and which have not.

LIBRARY (Cont.)

6.3.8 LIBRARY/REPLACE

LIBRARY/REPLACE replaces a module in a library with a new module of the same name and deletes the old module.

When a match occurs on a module name, the existing module is logically deleted and all its entries are removed from the global symbol table.

Format

LIBRARY/REPLACE libspec filespec[,s]

Command Qualifiers

/[NO]GLOBALS
/SELECTIVE_SEARCH
/SQUEEZE

Parameters

libspec

Specifies the library in which modules are to be replaced. The default file type is .OLB, corresponding to object libraries. If you wish to replace macros in a macro library (file type .MLB) or a universal library (file type .UNI), you must specify the file type explicitly.

filespec[,s]

Specifies the file or files containing the new modules. The default file type is determined by the type of library in which modules are being replaced. It is .OBJ for object module libraries, .MAC for macro libraries, and .UNI for universal libraries.

If the module to be replaced does not exist in the library, LIBRARY performs an insert operation.

If the module to be replaced exists, a message is printed. See example. If the module does not exist, it is inserted, but no message is printed.

Command Qualifiers

/[NO]GLOBALS

Specifies whether entry points for the specified modules should be included in the entry point table. The default is /GLOBALS.

/SELECTIVE_SEARCH

Sets the selective search attribute bit in the module header of object modules as they are inserted into an object library.

Object modules with the selective search attribute are given special treatment by the Task Builder. Global symbols defined in modules with the selective search attribute are only included in the Task Builder's symbol table if they were previously referenced by other modules.

LIBRARY (Cont.)**/SQUEEZE**

Reduces the size of macro definitions by eliminating all trailing blanks and tabs, blank lines, and comments from macro text.

Macros that have been squeezed not only take up less room in the macro library file but also take up less memory in the assembler when they are invoked.

See the Notes in Section 6.3 for more information on the /SQUEEZE qualifier.

This is the equivalent of the LBR /SZ switch applied to the input file.

Example

```
$ LIBR/REPLACE LB0:[1,1]SYSLIB USERSUB (RET)

MODULE "USERSUB" REPLACED
```

This example replaces the modules in the file SYSLIB.OLB with new modules bearing the same names in the file USERSUB.OBJ. Entry points are also redefined.

Notes

This command is the equivalent of the LBR /RP command.

There must be enough space in the library's tables for both the modules being replaced and their replacements since the new modules are entered before the old modules are deleted.

The old modules are logically deleted only. You must use LIBRARY/COMPRESS to eliminate the modules themselves and free the space they occupy.

6.3.9 Error Messages from LIBRARY

LIBRARY has two kinds of error messages, diagnostic and fatal.

Diagnostic messages are marked *DIAG* and inform you of conditions that require your consideration but do not warrant termination of the command. Depending on the message and the operation, you can decide what further action is called for. The operation completes.

Fatal messages are marked *FATAL* and describe conditions that caused LBR to terminate execution of your command. The operation does not complete.

In general, fatal output errors leave the library in an indeterminate state. You should use LIBRARY/LIST to determine how much of the operation completed before the fatal error occurred. If a library file is corrupted, the library must be re-created.

Some errors result from corrupted library files. You can use VFY, the File Structure Verification utility, to determine whether a file is in Files-11 format. A library file can be in Files-11 format and still be corrupted as a library, however.

LIBRARY (Cont.)

Error messages appear on your terminal in the following formats:

LBR -- *DIAG*-message

LBR--*FATAL*-message

Note that error messages from LIBRARY are tagged LBR after the utility and not LIB after the command. This is an exception to the general rule on DCL error messages.

LBR -- Bad library header

Explanation: The file either is not a library or is corrupted.

User Action: Make sure the file is a library. If the file is a library, use VFY, the File Structure Verification utility, to determine if it is corrupted. See Chapter 9 of the RSX-11M/M-PLUS Utilities Manual.

LBR -- Command I/O error

Explanation: There are two possible causes. The problem may be with the physical device, such as not being spun up. Or, the file may be corrupted or have an incorrect format, such as a record length exceeding 132 bytes.

User Action: Determine the condition that caused the message and correct the condition. Reenter the command line.

LBR -- Command syntax error
command line

Explanation: The reprinted command line has a syntax error.

User Action: Check for proper syntax and reenter the command line.

LBR -- Duplicate entry point name "name" in filename

Explanation: Your command attempted to insert (not replace) a module containing the given entry point name into a library already containing an entry point of that name.

User Action: Make sure you have named the correct input file. If not, reenter the command naming the correct input file. If the input file is correct, you can delete the duplicate entry point from the library and reenter the command.

LBR -- Duplicate module name "name" in filename

Explanation: Your command attempted to insert (not replace) a module with the given name into a library that already contains a module of that name.

User Action: Make sure you have named the correct input file. If the input file is correct, you must decide whether or not to use LIBRARY/REPLACE to replace the module that is already in the library.

LIBRARY (Cont.)

LBR -- EPT or MNT exceeded in filename

Explanation: The GLOBAL or MODULE table limit has been reached during the execution of a REPLACE or INSERT operation.

User Action: Copy the library using LIBRARY/COMPRESS and include the /MNT or /EPT qualifier to increase table size.

LBR -- EPT or MNT space exceeded in compress

Explanation: A GLOBAL or MODULE table size was specified in a COMPRESS operation for the output library file that is not large enough to contain the EPT or MNT entries used in the input library.

User Action: Reenter the command line with a larger EPT or MNT table size specified.

LBR -- Error in library tables, File "filename"

Explanation: Either the library is corrupted or it is not a library.

User Action: If the file is corrupted, no recovery is possible. The file must be reconstructed. If the file is not a library, reenter the command line with the correct file named.

LBR -- Fatal compress error

Explanation: The input library on a COMPRESS operation is corrupted or is not a library.

User Action: If the file is corrupted, no recovery is possible. The file must be reconstructed. If the file is not a library, reenter the command line with the correct file named.

LBR -- Get time failed

Explanation: System error. LBR attempted to execute a Get Time Parameters directive and failed.

User Action: Reenter the command line. If the error is repeated, see your system manager. Plan to submit a Software Performance Report. Save console dialogue and other pertinent information.

LBR -- Input error on filename

Explanation: The file system has detected an error.

User Action: Reenter the command. If the error is repeated, see your system manager.

LBR -- Insufficient dynamic memory to continue

Explanation: The partition in which LBR is running is too small.

User Action: See your system manager.

LIBRARY (Cont.)

LBR -- Invalid EPT and/or MNT specification

Explanation: Command included an argument for /MNT or /EPT greater than 4096.

User Action: Check for proper syntax and reenter command.

LBR -- Invalid module format, insertion-module

Explanation: Command attempted to insert the named macro definition into an object module library.

User Action: Correct error by reentering command with object module for object module library or macro for macro library.

LBR -- Invalid format, input file filename

Explanation: Your command named either an input file that is not in the standard format or a file that is corrupted.

User Action: Make sure you have named the right file and reenter command.

LBR -- Invalid library type specified

Explanation: Your command attempted to set an illegal library type as the default. The file types .OBJ and .MAC are the only valid default file types for the command.

User Action: Reenter command after checking proper syntax.

LBR -- I/O error on input file "filename"

Explanation: A read error occurred on an input file. This error can have two causes:

1. A problem exists with the physical device (for example, the disk may not be spun up).
2. The input file is corrupted or in the wrong format, such as records exceeding 132 bytes in length.

User Action: Determine the cause of the error and correct the condition. Reenter the command line.

LBR -- Library file specification missing

Explanation: Your command was entered without proper specification of the library file.

User Action: Check for proper syntax and reenter the command.

LIBRARY (Cont.)

LBR -- Mark for delete failure on LBR work file

Explanation: LBR automatically creates a work file when it begins processing commands and marks it for delete. For some reason this marking failed. The work file is a lost file and does not appear in any UFD.

User Action: See your operator or system manager. The file must be deleted using VFY, the File Structure Verification utility. See Chapter 9 of the RSX-11M/M-PLUS Utilities Manual.

LBR -- No entry point named "name"

Explanation: Your command named an entry point to be deleted that is not in the library.

User Action: Check command for proper syntax and reenter command. You may have misspelled the entry point name or named the wrong library.

LBR -- No module named "module"

Explanation: Your command named a module to be deleted that is not in the library.

User Action: Check command for proper syntax and reenter command. You may have misspelled the module name or named the wrong library.

LBR -- Open failure on file "filename"

Explanation: The file system detected an error while attempting to open a file. This error is caused by one of the following conditions:

1. The user directory area is protected against a Write operation.
2. There is a problem with the physical device, such as not being spun up.
3. The volume is not mounted.
4. The directory does not exist.
5. The file does not exist.
6. There is insufficient space to allocate the library file on a LIBRARY/CREATE or LIBRARY/COMPRESS.
7. There is insufficient pool space (DSR) in the Executive.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

LIBRARY (Cont.)

LBR -- Open failure on LBR work file

Explanation: The file system detected an error while attempting to open the LBR work file. The LBR work file is created on the volume from which LBR was installed. One of the following conditions can cause this error:

1. The volume is full.
2. The device on which the volume is mounted is write-protected.
3. There is a problem with the physical device.
4. There is insufficient pool space (DSR) in the Executive.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

LBR -- Output error on filename

Explanation: A write error has occurred on the output file. One of the following conditions can cause this error:

1. The volume is full.
2. The device on which the volume is mounted is write-protected.
3. The hardware has failed.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

LBR -- Positioning error on filename

Explanation: The device on which the volume is mounted is write-locked or some other hardware difficulty exists.

User Action: Determine the cause of the error if you can. If you cannot, see your operator or system manager.

LBR -- Virtual storage requirements exceed 65536 words

Explanation: This error may occur with maximum size libraries in conjunction with a single command line that logically deletes a large number of modules and entry points and continues to replace them with an equally large number of modules and entry points having highly dissimilar names. Normally, this message indicates some sort of internal system error.

User Action: Rerun the job after dividing the command line into several smaller lines that perform the same operations.

LIBRARY (Cont.)

LBR -- Work file I/O error

Explanation: A write error has occurred on the LBR work file. One of the following conditions can cause this error:

1. The volume is full.
2. The device on which the volume is mounted is write-protected.
3. The hardware has failed.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

6.4 SET AND SHOW

You can set and show group global event flags. These are a set of 32 global event flags available only to tasks run from one group of UICs. Otherwise, these flags are identical to other global event flags. Event flags are used for communication between tasks. See the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual for further information.

SET GROUPFLAGS

6.4.1 SET GROUPFLAGS

SET GROUPFLAGS creates and deletes group global event flags. Privileged users can create and delete group global event flags for any group. Nonprivileged users can create and delete global event flags for the group of which they are members.

Format

```
SET GROUPFLAGS[/qualifier]
Flag? n

SET GROUPFLAGS:n[/qualifier]
```

Command Qualifiers

```
/CREATE
/DELETE
```

Command Qualifiers

/CREATE

Specifies that you wish to create a set of group global event flags. This is the default and need not be specified.

The argument n is the group with which the flags are associated. Nonprivileged users can create and delete group global event flags for their own login group. Privileged users can create and delete group global event flags without restriction.

/DELETE

Specifies that you wish to delete a set of group global event flags. The argument n is the group with which the flags are associated.

Example

```
$ SHOW GROUPFLAGS (RET)
7      0      000000
200    0      000000
$
$ SET GROUPFLAG (RET)
Flag? 303 (RET)
$
$ SHOW GROUPFLAG (RET)
7      0      000000
200    0      000000
303    0      000000
```

In this example, the user first displayed all the current group global event flags on the system and found that there were two sets but none for the user's group, which was 303. Then the user created a set of group global event flags for his group and verified their presence with the SHOW GROUPFLAGS command.

SHOW GROUPFLAGS**6.4.2 SHOW GROUPFLAGS**

SHOW GROUPFLAGS displays the group global event flags currently in the system.

Format

```
SHOW GROUPFLAGS
```

Example

```
$ SHOW GROUPFLAGS (RET)
7      1      000000 000000
200    1      000000 000000
201    1      000000 000000
303    1      000000 000000
333    1      000000 000000
```

This example shows the display from SHOW GROUPFLAGS. The first column is the group number with which the flags are associated. The second column is the access count. The two 6-digit numbers are the octal words that give the current state of each group global event flag.

Group global event flags are flags 65 through 96. The first word starts at flag 80 and goes right to flag 65. The second word starts at flag 96 and goes right to flag 81. The final column is reserved for the delete flag DEL, which means the group global event flags are marked for delete and are not available. See the previous section.

CHAPTER 7

RUNNING TASKS

The task is the fundamental executable programming unit.

Tasks are built (linked) from one or more object modules. Tasks on RSX-11M systems include tasks built from user programs as well as system tasks, such as compilers or utilities.

Commands described in this chapter are used to initiate and terminate execution of tasks and to display the status of tasks.

7.1 TASK INSTALLATION AND EXECUTION

Tasks cannot execute unless they are installed in the system. The System Task Directory (STD), a data structure in the Executive's dynamic storage region (pool), is the list of installed tasks. Each installed task is identified by a Task Control Block (TCB) in the STD. You can display the contents of the STD with SHOW TASKS/INSTALLED. See Section 7.12.3.2.

Nonprivileged users can install their own tasks through the RUN command. These tasks remain installed as long as they are executing, and are removed from the system when they finish. Privileged users can install tasks. These tasks can also be caused to execute with the RUN command. The tasks remain installed after execution.

Many system tasks can be caused to execute with other commands. For instance, EDIT/EDT causes EDT, DIGITAL's interactive editor, to execute on your terminal, and LINK causes the Task Builder to execute.

Installed tasks are not necessarily resident in memory or competing for system resources. Most installed tasks are dormant, meaning they are installed but have not been requested to run. A dormant task uses little memory, yet it is immediately available when it is needed, such as to service a real-time event.

Installed tasks can be in either of two states:

Dormant	Installed but not requested to run
Active	Installed and requested to run

Tasks remain active until they abort, exit, or terminate in some other way. During this time, they can be in one of two substates:

Ready-to-run	Competing for CPU time on the basis of priority
Blocked	Unable to compete for CPU time either because a needed resource is not available, or for synchronization purposes, or through the STOP/BLOCK command (Section 7.6).

RUNNING TASKS

You can display the list of active tasks with SHOW TASKS/ACTIVE. See Section 7.12.3.1.

Tasks run at priorities from 1 through 250, with 250 being the highest priority. The priority of a task can be established through the LINK, INSTALL, or RUN commands. Nonprivileged users are limited to priority 50 in the commands they issue, but if a privileged user has built or installed the task to run at a higher priority, then it will run at that priority for both privileged and nonprivileged users.

Tasks can be built, installed, or run as checkpointable. Checkpointable tasks are liable to being checkpointed, or rolled out of memory to disks, in an incomplete state if a higher-priority task needs the memory space they are occupying.

All memory is divided into partitions, which are subdivisions devoted to a particular task or tasks or to system functions. Partitions can be dedicated to a single task or shared by several tasks. If you do not specify a partition when you install and run a task, it will be installed in the default partition, named GEN. All partitions have a name and a size.

Refer to the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information on how tasks are built, installed, and run in partitions.

7.1.1 Task Naming

Tasks are installed under names from one through six Radix-50 characters. The names identify the task in the STD. All commands and other operations affecting installed tasks require you to use the name of the installed task. This section describes a number of ways tasks acquire their installed names.

Many task names include the number of the terminal from which the task was initiated. Terminal numbers are octal. Unless otherwise stated, all commands in examples in this section are assumed to have been issued from terminal TT10:.

Tasks resulting from system commands are named after the command and the terminal from which the command was issued. Thus, the following command, issued from terminal TT10:

```
$ SHOW TIME (RET)
```

results in a task named SHOT10 in the STD while the task is executing.

And, the following command, passed through DECnet by host terminal number 6:

```
SHOW TIME
```

results in a task name SHOH6 while the task is executing. (A host terminal is a DECnet virtual terminal.)

For tasks that are installed and run explicitly, task names can be established through the LINK command, the INSTALL command, and the RUN command. For example, if you begin with a source file named ROBOT.MAC and use the default LINK command after assembling the object file, you will end up with a task image file named ROBOT.TSK. You can install this task image in two ways, through the privileged INSTALL command or through the nonprivileged RUN command.

RUNNING TASKS

If you issue the following privileged command:

```
$ INSTALL ROBOT.TSK (RET)
```

you will find the task name ROBOT in the STD.

If you issue the following command:

```
$ RUN ROBOT.TSK (RET)
```

you will find the task name TT10 in the STD while the task is executing. The installation is through the install-run-remove form of RUN, which assigns a task name based on the name of the terminal from which the command was issued.

You can override these defaults at every step.

The TASK= option of LINK permits you to specify the name under which you wish the task image file you are creating to be installed. If you thus specify the installed name of ROBOT1.TSK to be EANDO when you build the task, the following privileged command:

```
$ INSTALL ROBOT1.TSK (RET)
```

results in the task name EANDO in the STD.

If, however, you issue the following command:

```
$ RUN ROBOT1.TSK (RET)
```

you will again find the task name TT10 in the STD while the task is executing, despite the specification of EANDO at task build time.

The following command:

```
$ RUN/TASK_NAME:EANDO ROBOT1.TSK (RET)
```

results in the task name EANDO in the STD while the task is executing.

The RUN command defaults to a task name that is based on the terminal designation (without the colon). Therefore, you must use the /TASK NAME qualifier if you want to use the RUN command to initiate execution of more than one task at a time from the same terminal. This is true because all task names in the STD are unique; you cannot have two tasks named TT10 installed at the same time, for example. See the examples in Section 7.2. Finally, most DCL and MCR commands work by running tasks, although you do not have to issue a RUN command. The LINK command, for instance, runs the Task Builder. Such tasks are named after the first three letters of the command word, plus the terminal identifier. Thus, when you issue a command such as the following:

```
$ LINK RUMBLE (RET)
```

the result is a task named LINT10. In fact, the RUN command itself results in the brief existence of a task named RUNT10.

If you are running a utility, using a command in the following form:

```
$ MCR PIP (RET)
```

your task will have a name of the form PIPT10.

See Section 7.2 for more information on including the dollar-sign (\$) in a RUN command.

RUNNING TASKS

On RSX-11M systems, there is a possibility of a slight variation in the task-naming rules. If, when you run a utility, you are the only user using that utility, the utility is installed under its prototype name, which has a name preceded by three dots, such as ...PIP. Any subsequent users run the utility under a name in the form PIPT10. It makes no difference to you whether you are running the prototype or a copy.

Only uninstalled tasks initiated with the RUN command have task names preceded by TT, or HT. Tasks initiated by other DCL commands are named after the first three letters of the command that initiated the task, plus a T or H (for physical, or DECnet host terminal) plus the unit number.

RUN

7.2 INTRODUCTION TO THE RUN COMMAND

RUN initiates the execution of a task. This command can be used in four ways:

1. To install, run, and remove upon execution a task from a task image file stored in a user's UFD. This is the most common use of the command.
2. To install, run, and remove after execution a task from a task image file stored in the system UFD.
3. To run immediately a task previously installed by a privileged user.
4. To run at some future time, or according to a schedule, a task previously installed by a privileged user.

These functions are introduced in more detail in the following paragraphs.

The first and most common use of RUN is to initiate execution of tasks contained in task image files. The following command:

```
$ RUN PUPPET.TSK (RET)
```

directs the operating system to look in the default UFD on the default device for a task image file named PUPPET.TSK. When the file is found, the image is read in to memory and execution commences. When the task has executed, the image is removed from memory and from the STD.

Whenever a RUN command includes any element of a file specification, this procedure is followed. The following command:

```
$ RUN DB0:MUPPET (RET)
```

directs the operating system to do the following:

- Look in the default directory on device DB0: for a file named MUPPET.TSK.
- Install and run the task image.
- Remove the task image when it has completed execution.

The default file type in this case is .TSK, which is also the default file type for a task image file created by the Task Builder.

The second use of RUN is to initiate execution of tasks contained in task image files in a system directory. The dollar sign (\$) in a file specification indicates that you want to use this form of RUN. The purpose of this convention is to relieve you of the necessity of knowing exactly where on the system various system tasks are stored.

RUN (Cont.)

Simply include the dollar sign (\$) in your RUN command and the system takes over. If you receive the following error message:

```
RUN -- File not found
```

or

```
INS -- File not found
```

your system manager has chosen not to include that particular task or utility in this directory, perhaps to conserve disk space. In this case, see your system manager.

In general, you can use the dollar sign to run most system tasks and utilities. These usually have 3-letter names like PIP or TKB, but the RMS-11 utilities have 6-letter names like RMSDSP or RMSCNV. See the RMS-11 documentation supplied with your system for more information on RMS-11. (Your system manager may also have added other task image files not supplied by DIGITAL to this directory for your convenience.)

On RSX-11M systems, copies of many or all system tasks are placed in the system directory on pseudo device LB:. The directory is usually [1,54], but it can be some other number. You can display the identity of this directory with the command SHOW SYSTEM/DIRECTORY. You can list the contents of the directory with a DIRECTORY command.

The system responds to the dollar sign (\$) in a file specification as follows. The following command:

```
$ RUN $TKB (RET)
```

directs the operating system to do the following:

- Look for a file called TKB.TSK.
- Install and run the task image.
- Remove the task image when it has completed execution.

Because the Task Builder is a privileged task, it will be found in the system directory. The system installs, runs, and removes TKB just like any other task.

NOTE

Privileged users can also use the dollar sign (\$) with the INSTALL command. See Section 7.8.

The third use of RUN is to execute immediately a previously installed task. The task must have been installed by a privileged user. The following command:

```
$ RUN TURTLE (RET)
```

directs the operating system to look in the list of installed tasks (STD) for a task named TURTLE. Once the system has found the task TURTLE in the STD, the task commences its execution. The task need not be installed since it is already installed, nor is it removed when it has completed its execution. The task runs under the name TURTLE, not TT10.

RUN (Cont.)

(If the system does not find an installed task of the name you give, the command defaults to the install-run-remove form and looks in the default UFD on the default device for a file named TURTLE.TSK, and continues as described earlier.)

Finally, privileged users can use RUN to schedule the execution of a task in several forms. Scheduling options include:

- Run the installed task after so many hours, minutes, seconds, or ticks.
- Run the installed task at an absolute time of day.
- Run the installed task in synchronization with the next hour, minute, second, or tick, that is, at the beginning of the next interval.
- Run the installed task at regular intervals.

In addition, these scheduling options can be combined.

RUN as used to install, run, and remove tasks contained in task image files is described in Section 7.2.1. RUN as used with previously installed tasks is described in Section 7.2.2.

7.2.1 Running Tasks Contained in Task Image Files

When used to run an uninstalled task, RUN is actually a combination command, encompassing INSTALL, RUN, and REMOVE. This command is the only way a nonprivileged user can install or remove a task.

Format

```
RUN[/qualifier[s]]
Task? [$]filespec
```

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

Command Qualifiers

```
/[NO]CHECKPOINT
/COMMAND:"taskcommand"
/EXTENSION:n
/[NO]IO_PAGE
/PARTITION:parname
/[NO]POSTMORTEM
/PRIORITY:n
/[NO]SLAVE
/STATUS:arg
      TASK
      COMMAND
/TASK NAME:taskname
/UIC:[g,m]
```


RUN (Cont.)

Parameter

[\$]filespec

Specifies a task image file on a mounted Files-11 volume. This task is installed, run, and then removed when it has finished executing.

The default file type is .TSK. See the Notes.

The dollar sign (\$) directs the system to search first for the file in the system directory. In this case, the dollar sign specifies a device and directory and thus counts as an element of a file specification.

Command Qualifiers

These qualifiers establish how the uninstalled task is to be installed.

/[NO]CHECKPOINT

Specifies whether or not the task is to be run checkpointable. The default is established when the task is built (linked). This qualifier is used to override the default.

In other words, this qualifier allows you to run checkpointable tasks as noncheckpointable and vice versa.

/COMMAND:"taskcommand"

Passes the specified command line to the task you are running. The command must begin with the task's acronym and it must be valid for the task or a syntax error will occur. Also, the command must be inside the quotation marks (") and not more than 40 characters long. For example, the following command:

```
$ RUN/COMMAND:"PIP /LI" $PIP (RET)
```

runs PIP and then passes the /LI switch to it. This results in a directory listing.

When the task has finished executing the command line, the task is removed.

/EXTENSION:n

Specifies that n additional decimal words of address space be allocated to a task.

/[NO]IO_PAGE

Specifies that a privileged task can overmap the I/O page. If you specify /IO_PAGE, RUN assumes that the task will overmap the I/O page and, if the task is larger than 8K words, issues a warning message. If you specify /NOIO_PAGE, RUN assumes that the task does not need to use the I/O page; no warning message is issued. The default is /IO_PAGE.

RUN (Cont.)

/PARTITION:parname

Specifies the partition in which the task is to run. The default is established when the task is built (linked). This qualifier is used to override the default.

If no partition is established at task build or with the RUN command, the task will run in the default partition GEN.

/[NO]POSTMORTEM

Specifies whether or not a Postmortem Dump is to be generated if the task terminates unexpectedly.

The default is determined when the task is built. If not specified, the LINK command defaults to /NOPOSTMORTEM. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information on Postmortem Dumps.

/PRIORITY:n

Specifies at what priority the task is to run. This is a privileged qualifier. The default is established when the task is built (linked). This qualifier is used to override the default.

The default priority for the Task Builder, INSTALL, and RUN is 50 (decimal). If a task has been built or installed at another priority, nonprivileged users can run it at that priority. Otherwise, nonprivileged users are limited to running tasks at the default priority.

Priorities range from 1 through 250 (decimal).

/[NO]SLAVE

Specifies the slave status of a task. If you specify /SLAVE, data sent to the installed task is marked with the TI: of the sending task. When the installed task receives the data, the system sets the task's TI: to that associated with the data. This qualifier overrides the LINK qualifier /SLAVE specified when the task was built. The default is specified at task-build time.

/STATUS:arg

TASK
COMMAND

Specifies whether exit status is to be returned from the RUN command or from the task being run with the RUN command. The default is /STATUS:COMMAND and need not be specified.

The /STATUS:TASK qualifier may be necessary in indirect command files that run installed tasks where the indirect command processor must wait for the installed task to exit before attempting to execute the next command.

RUN (Cont.)

`/TASK_NAME:taskname`

Specifies the name under which the task is to be run. The default is to run the task under a name derived from the name of the terminal from which the RUN command was issued, as discussed in Section 7.1.1.

Use this qualifier when you wish to run two tasks simultaneously using the RUN command. Otherwise, the second RUN command causes the following error message to appear on your terminal:

```
RUN -- Task name already in use
```

See the examples.

Task names are restricted to six Radix-50 characters. The Radix-50 character set consists of the 26 uppercase letters, the 10 numerals, the period (.), and the dollar sign (\$).

`/UIC:[g,m]`

Specifies the default UIC for the task. This is a privileged qualifier. This UIC determines in what protection class the task belongs, and thus determines file access.

The square brackets are required syntax.

Examples

```
$ RUN (RET)
Task? ROBOT (RET)
```

This example installs, runs, and removes upon completion of execution the task contained in the task image file named ROBOT.TSK from the default directory on the default device.

```
$ RUN ROBOT (RET)
$ RUN ROBOT.TSK (RET)
```

The two commands in this example are the equivalent of the command in the previous example, assuming there is no installed task named ROBOT. If there is, only the second form, with the file type specified, will run the task from the directory.

The following examples illustrate the task-naming convention. See Section 7.1.1 for more information on task naming.

```
$ RUN/TASK_NAME:CAPEK ROBOT (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR...
SHOT10
CAPEK
```

This example shows how the task-naming convention can be overridden for the RUN command by running the task under another name using the `/TASK_NAME` qualifier.

RUN (Cont.)

```

$ RUN ROBOT (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR...
SHOT10
TT10
$ RUN ASIMOV (RET)
RUN -- Task name already in use
$ RUN/TASK:BINDER ASIMOV (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR...
SHOT10:
TT10
BINDER
    
```

This example shows how you can run two tasks simultaneously using the RUN command and the /TASK_NAME qualifier.

Notes

The return of the dollar sign prompt (\$) signifies that the task initiated by the RUN command has exited. However, you need not wait for the prompt to return to issue another command. Simply press RETURN or CTRL/C and enter the command. If you specified the /STATUS:COMMAND qualifier, the dollar sign prompt returns immediately after you issue a RUN command for an uninstalled task, but does not indicate that the task has completed its execution.

This section describes RUN as it works with any portion of a file specification included. In general, you do not have to include the file type, as .TSK is the default. In one unusual circumstance, however, you must explicitly enter a file type. If there is a task already installed in the system that has the same task name as the file name of the task image file containing the task you wish to run, the RUN command will run the installed task instead of the one you wish to run. In such an instance, you must include the file type or some other part of the file specification, and include the dollar sign, to force the RUN command to access the task image file.

Some error messages resulting from RUN are labeled INS rather than RUN because this form of RUN installs the task automatically and the error occurs during the installation.

7.2.2 Running and Scheduling Installed Tasks

RUN also initiates the execution of installed tasks. Privileged users can use RUN to initiate the execution of installed tasks on a schedule by creating entries in the system clock queue.

Tasks that run as a result of entries in the clock queue run with pseudo device C00: as their TI:. For this reason, these commands are privileged. Such tasks send output to C00: and not to the terminal they were run from.

RUN (Cont.)

Format

```
RUN[/qualifier[s]]
Task? taskname
```

```
RUN[/qualifier[s]] taskname
```

Command Qualifiers

```
/DELAY:nu
/INTERVAL:nu
/SCHEDULE:hh:mm:ss
/STATUS:arg
        COMMAND
        TASK
/SYNCHRONIZE:u
/UIC:[g,m]
```

Parameter

taskname

Specifies the name of the installed task to be run. The names of installed tasks are one through six Radix-50 characters. If the System Task Directory (STD) contains no entry under the task name you supply, the system searches the default directory on the default device for a file named taskname.TSK. If RUN finds such a file, it installs, runs, and removes it.

Command Qualifiers

The time-oriented qualifiers to RUN create entries in the system clock queue. The contents of the clock queue can be displayed with the command SHOW CLOCK_QUEUE. See Section 7.12.5.

/DELAY:nu

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

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

If your system has the standard line-frequency clock, the length of a tick depends on the line frequency of the electric power source serving your location. Most commonly, this frequency is 60 Hz, which results in a tick length of 1/60th of a second.

If your system has an optional programmable clock, the length of a tick depends on the choice made at system-generation time. See your system manager for more information.

Acceptable values for these units are as follows:

```
T - Any positive value to a maximum of 15 bits, or 32767.
S - Any positive value to a maximum of 15 bits, or 32767.
```

RUN (Cont.)

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. This qualifier is privileged. The argument nu specifies the interval as a number of units of time. 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. This is a privileged qualifier.

/STATUS:arg

COMMAND
 TASK

Specifies whether exit status is to be returned from the RUN command or from the installed task being run with the RUN command. This is not a privileged qualifier. The default is /STATUS:COMMAND and need not be specified.

The /STATUS:TASK qualifier may be necessary in indirect command files that run installed tasks, where the indirect command processor must wait for the installed task to exit before attempting to execute the next command.

This qualifier can be specified separately or with the /UIC qualifier.

/SYNCHRONIZE:u

Specifies that the execution of the task be synchronized on the next occurrence of a particular clock unit. This is a privileged qualifier. The argument u is the time unit as follows:

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

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

RUN (Cont.)

/UIC: [g,m]

Specifies the default UIC for the task. This is a privileged qualifier. The task's UIC determines what file-protection class it belongs in and thus directly influences file access.

The square brackets are required syntax.

Examples

```
$ RUN/DELAY:5S PINBAL (RET)
```

This example initiates execution of the installed task PINBAL five seconds after the command is entered.

```
$ RUN/INTERVAL:20M WIZARD (RET)
```

This example initiates execution of the installed task WIZARD immediately and every 20 minutes thereafter.

```
$ RUN/DELAY:10M/INTERVAL:1H TOMMY (RET)
```

This example initiates execution of the installed task TOMMY after ten minutes and every hour thereafter.

```
$ RUN/SCHEDULE:12:00:00 NOON (RET)
```

This example initiates execution of the installed task NOON at the next 12:00:00 after the command is entered. If you entered the command before noon today, NOON would run at noon today. If you entered the command after noon today, NOON would run at noon tomorrow.

```
$ RUN/SYNCHRONIZE:H BOGUS (RET)
```

This example initiates execution of the installed task named BOGUS on the next hour. If you entered the command at 9:15, BOGUS would begin running at 10:00:00.

```
$ RUN/SYNCHRONIZE:H/INTERVAL:1H BELTOL (RET)
```

This example initiates execution of the installed task named BELTOL every hour on the hour, starting with the next hour.

Notes

The dollar sign prompt (\$) prompt returns immediately after you issue a RUN command for an installed task and does not indicate that the task has completed its execution.

You can display entries in the clock queue with SHOW CLOCK_QUEUE, Section 7.12.5.

The clock queue is kept in pool and must be restored whenever the system is rebooted.

RUN (Cont.)

Error Message

RUN -- Invalid time parameter

Explanation: Command specified an invalid time argument.

User Action: Check for proper syntax and reenter command.

ABORT

7.3 ABORT

ABORT forces an orderly end to a running task or to the action of a specific command.

Tasks can also be aborted by other tasks or by error-handling routines. If this happens, some of the messages discussed below can appear on your terminal without your having issued an ABORT command.

Nonprivileged users can abort any task running on TI:. Privileged users can abort any task.

Format

```
ABORT[/COMMAND][qualifier[s]] commandname
```

```
ABORT/TASK[/qualifier[s]] [taskname]
```

Command Qualifiers

```
/COMMAND
```

```
/TASK
```

```
/[NO]POSTMORTEM
```

Parameters

The parameters differ depending on whether you are aborting a command or a task. The default is to abort a command. See the examples.

commandname

Specifies the command whose effect you wish to cancel. This parameter can be used only when the /TASK qualifier is not present. You must specify at least the first three characters of the command verb.

taskname

Specifies the name of the task you wish to abort. This parameter requires the presence of the /TASK qualifier. If you use the /TASK qualifier and do not specify a task name, you will get an error message, "Illegal task name."

Command Qualifiers

```
/COMMAND
```

Specifies that you wish to abort a command. This is the default qualifier and need not be included.

```
/[NO]POSTMORTEM
```

Specifies that a Postmortem Dump be taken (or not taken) of the task before it is aborted. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information.

ABORT (Cont.)

If the task is memory-resident, the contents of its registers are listed on the terminal. If it is checkpointed, no register contents are listed.

You can specify a Postmortem Dump when you build a task or when you install it (including install-run-remove). The default for this qualifier is determined by previous specification of the /DUMP qualifier. If no /POSTMORTEM qualifier has been specified, the default is /NOPOSTMORTEM.

/TASK

Specifies that you want to abort a task by name.

Examples

```
$ ABO RUN (RET)
11:11:11 Task "TT10" terminated
        Aborted via directive OR CLI
```

This example aborts a task initiated with the RUN command running on your terminal. See next example.

```
$ RUN TREK (RET)
ORDERS: STAR DATE = 2000
YOU MUST DESTROY THE KLINGON INVASION FORCES OF 29 BATTLE
CRUISERS. YOU HAVE 40 SOLAR YEARS TO COMPLETE YOUR MISSION.
READY? (CTRL/C)
DCL>ABORT RUN (RET)
12:11:09 Task "TT10" terminated
        Aborted via directive or CLI
```

In this example, the user installed the interactive task TREK by means of a RUN command. Then, instead of providing the task with requested input, the user issued a CTRL/C to return to the DCL monitor level and then issued the ABORT command. DCL prompted for the command to be aborted and RUN was specified. The task was aborted and removed, and the message displayed. The message mentions MCR because DCL does not actually perform the abort.

```
$ ABORT/TASK MACT3 (RET)
```

This example aborts the task named MACT3. Nonprivileged users can abort tasks running on their own terminals. Privileged users can abort tasks by name running on any terminal. See next example.

ABORT (Cont.)

\$ DIRECTORY *.RNO (RET)

Directory DB2: [303,5]
20-MAR-85 16:28

ALTCLI.RNO;3	213.	27-AUG-84 16:52
ANNI.RNO;6	3.	30-JUL-84 09:30
APRCOM.RNO;4	9.	15-FEB-85 10:35
BEST.RNO;5	40.	28-AUG-84 15:50
CATCH.RNO;4	7.	12-MAR-85 09:39
CHAP11SMG.RNO;13	49.	13-OCT-84 16:36
CHAP12SMG.RNO;51	223.	08-OCT-84 13:15
CHARSET.RNO;5	6.	15-JAN-85 11:57 (CTRL/C)

DCL>ABORT DIRECTORY (RET)
16:29:45 Task "DIRT56" terminated
Aborted via directive or CLI

\$

In this example, the user issued a DIRECTORY command and soon saw that the file he was checking for was present. Rather than wait for the directory listing to complete, the user typed CTRL/C and entered the ABORT command in response to the dollar sign (\$) prompt. The ABORT command defaults to ABORT/COMMAND.

\$ ABORT/DUMP RUN (RET)

14:41:11 Task "TT10 " terminated
Aborted via directive or CLI
R0=000000
R1=100077
R2=135600
R3=000000
R4=000000
R5=000000
SP=012540
PS=170017

\$

In this example, the user requested a Postmortem Dump. The contents of the task's registers at the time of the abort are listed on the terminal along with notification of the dump. The dump file will be printed on the system's line printer.

Notes

A is the short form of ABORT.

ABORT forces an orderly termination of a task. To effect the termination, the system performs four steps:

1. Alters the task's priority to 247. A task must be memory resident to be aborted. This step usually forces the quick return of a checkpointed task.

ABORT (Cont.)

2. Performs I/O rundown: cancels I/O for nonfile-structured devices; completes I/O for file-structured devices and deaccesses files; detaches all attached devices.
3. Executes the Task Termination Notification program (TKTN), which displays messages on the aborted task's TI:.
4. Releases any memory allocated to the task if the task is not fixed.

The TKTN routine can list a number of messages about the abort. These messages are explained below.

LOGOUT also aborts nonprivileged tasks running from your terminal.

Tasks can also be aborted by Executive directives issued by other tasks, in particular, by error-handling routines.

TKTN Messages

These messages appear as part of the notification of the abort. Often a list of register contents also is displayed. Understanding these messages requires an understanding of how the PDP-11 executes instructions. If you receive these messages and do not understand them, you should copy them, along with any other terminal output, and show them to your system manager.

Aborted via directive or CLI

Explanation: Either a CLI or an Executive directive issued by another task caused the task to be aborted.

AST abort. Bad stack

Explanation: An AST could not be effected because the AST parameters could not be pushed on the stack.

Checkpoint failure. Read error

Explanation: The task could not be read back into memory after being checkpointed.

IOT execution

Explanation: The task executed an IOT instruction, and no SST routine was specified to process the trap.

Load failure. Read error

Explanation: The task could not be loaded because of a hardware error.

Memory protect violation

Explanation: The task encountered a memory-protect violation and no SST routine was specified to process the trap.

ABORT (Cont.)

Non RSX EMT execution

Explanation: The task executed an EMT instruction with an argument other than 377(octal) or, in the case of a privileged task switching to system state, 376(octal), and no SST routine was specified to process the trap.

Odd address or other trap four

Explanation: The task executed a word instruction with an odd address, or it referenced a nonexistent memory location, and no SST routine was specified to process the trap.

Parity error

Explanation: A parity error occurred while the task was executing. The task was fixed in memory so that the memory could not be reused by another task.

Reserved INST instruction

Explanation: The task executed an illegal instruction, and no SST routine was specified to process the trap.

SST abort. Bad stack

Explanation: An SST could not be effected because the SST parameters could not be pushed onto the task's stack.

Task exit with outstanding IO

Explanation: The task exited with one or more outstanding I/O requests. Tasks should terminate all I/O operations before exiting. The system does, however, clean up all outstanding I/O.

T bit trap or BPT instruction

Explanation: The task either set the T bit in the Processor Status Word or executed a BPT instruction, and no SST routine was specified to process the trap.

Trap execution

Explanation: The task executed a Trap instruction, and no SST routine was specified to process the trap.

Error Messages

ABO -- Task marked for abort

Explanation: A previous command requested that the task be aborted, but the abort has not yet completed. This may mean the task marked for abort is not currently resident in memory.

User Action: None.

ABO -- Task not active

Explanation: The specified task is not currently active.

User Action: Check command for proper task name and reenter it.

CONTINUE

7.4 CONTINUE

CONTINUE resumes execution of a previously suspended task.

Nonprivileged users can continue tasks initiated from their own terminals. Privileged users can continue any suspended task.

Format

CONTINUE [taskname]

Parameter

If you do not supply a parameter, the task running at your terminal initiated by the RUN command is continued.

Tasks are suspended through the execution of a SPND\$\$ directive within the task. A RSUM\$ directive is another way of continuing a suspended task.

taskname

Specifies the suspended task you want to continue. If you do not specify a task name, the default is a task named after the terminal.

Example

```

$ RUN YOYO (RET)
.
.
.
"Message from Yoyo: Yoyo is suspended"
$ SHOW TASK/ACTIVE TT10 (RET)
TT10 070530 GEN 00677100-00701600 PRI - 50. DPRI - 50.
STATUS: -CHK SPN -PMD REM MCR
TI - TT10: IOC - 0. EFLG 000001 000000 PS 170000 PC 001254
REGS 0-6 001372 000037 000012 140311 001254
$ CONTINUE (RET)
"Message from Yoyo: Not suspended. Climbing string"
$
    
```

In this example, the user ran a task that issues a message and then suspends itself. Not all suspended tasks issue such messages, but you can identify a suspended task through SHOW TASKS/ACTIVE. The user issued SHOW TASKS/ACTIVE and named the task. Included in the listing under STATUS on the second line the mnemonic SPN verifies that the task was suspended. The user issued CONTINUE without naming the task. Since the task was initiated by RUN, CONTINUE defaulted to a task name based on the name of the terminal.

Notes

A task cannot suspend any task other than itself.

For more information on task states, see the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual.

CONTINUE (Cont.)

Error Message

CON -- Task not suspended

Explanation: Command attempted to continue an unsuspended task.

User Action: None.

CANCEL

7.5 CANCEL

CANCEL eliminates entries from the clock queue. Entries are placed in the clock queue through the RUN\$ directive or from the privileged time-based forms of the RUN command.

Nonprivileged users can only cancel entries from a task initiated from the entering terminal. Privileged users can cancel any clock queue entries.

CANCEL does not affect a currently executing task, but only the pending entries in the clock queue.

Format

```
CANCEL
Task? taskname
```

```
CANCEL taskname
```

Parameter

taskname

Specifies the name of the installed task whose clock-queue entries you wish to eliminate.

Example

```
$ RUN/DELAY:5M ARDVRK (RET)
$ SHOW CLOCK_QUEUE (RET)
ARDVRK Scheduled at 17-APR-85 10:44:31:09
$ CANCEL ARDVRK (RET)
$ SHOW CLOCK_QUEUE (RET)
$
```

In this example, the user issued a command to run the installed task ARDVRK at exactly five minutes from when the command was entered. The user verified that ARDVRK was in the clock queue (as the only entry). Then the user issued the CANCEL command to eliminate the scheduled run of ARDVRK. The user issued a second SHOW CLOCK_QUEUE to verify that the scheduled entry was cleared. If ARDVRK had been running at the time, the CANCEL would have had no effect.

Notes

See Section 7.12.5 for information on SHOW CLOCK_QUEUE.

STOP/BLOCK

7.6 STOP/BLOCK

STOP/BLOCK blocks an installed running task. The task no longer executes or competes for memory. Nonprivileged users can block tasks running from their own terminals. Privileged users can block any task.

Format

```
STOP/BLOCK [taskname]
```

Parameter

taskname

Names the task to be blocked. If you do not name a task, the task initiated with the RUN command and named after the terminal is blocked.

Example

```
$ RUN ADVENT (RET)
$ SHO TASK/ACTIVE (RET)
MCR...
SHOT5
TT5
$ STOP/BLOCK (RET)
$ SHOW TASK:TT5/FULL (RET)
TT5      100604  GEN      073464 00541000-005436 PRI - 50. DPRI -
50.
STATUS:  BLK -CHK WFR -PMD REM MCR
TI - TT5:  IOC - 0.  BIO - 0.  EFLG - 000001 000000 PS - 170000
PC - 002206 REGS 0-6 001637 040066 040060 140311 163500 000000 001252
$
$ START/UNBLOCK (RET)
$ SHOW TASK:TT5/FULL (RET)
TT5      100604  GEN      073464 00541000-005436 PRI - 50. DPRI - 50.
STATUS:  -CHK WFR -PMD REM MCR
TI - TT5:  IOC - 1.  BIO - 0.  EFLG - 000001 000000 PS - 170000
PC - 002206 REGS 0-6 006173 042066 040060 140311 163500 000000 001260
$
```

In this example, the user on terminal TT5: ran a task with the RUN command, giving the task the name TT5 by default. The user then issued the STOP/BLOCK command, without naming the task. The /FULL display from SHOW TASKS shows task TT5's status as blocked (BLK flag). Then the user issued the START/UNBLOCK command and checked the status again. This time, the BLK flag is not present.

Notes

The START/UNBLOCK and ABORT commands are the only means of canceling the STOP/BLOCK command.

A blocked task has an effective priority of 0. If it is checkpointable, any task can checkpoint it. Once checkpointed, the blocked task stays checkpointed because it is ineligible to compete for memory. A blocked, checkpointed task is unblocked when it is aborted.

STOP/BLOCK (Cont.)

A task can also be blocked by the system while waiting for some needed resource or for synchronization purposes.

There are no prompts from this command.

See Section 7.12.3 for an explanation of the task status flags.

There are several STOP/function commands associated with the Queue Manager. See the RSX-11M/M-PLUS System Management Guide.

START

7.7 START

The START command resumes execution of a task that was stopped by the STOP\$\$ directive. Note that starting a stopped task is different from continuing a suspended one.

Format

```
START[/TERMINAL:ttnn:] [taskname]
```

Parameter

taskname

Specifies the name of the task you wish to start. If you do not specify a taskname, the task running at your terminal that was initiated by the RUN command is started, that is, a task named after your terminal.

Command Qualifier

/TERMINAL:ttnn:

Allows a privileged user to start a stopped task that is running on some other terminal.

Notes

Several other forms of the start command are used to control the Queue Manager. See the RSX-11M/M-PLUS Batch and Queue Operations Manual and the RSX-11M/M-PLUS System Management Guide for more information.

START/UNBLOCK

7.7.1 START/UNBLOCK

START/UNBLOCK continues the execution of a task blocked by the STOP/BLOCK command. Nonprivileged users can unblock any task running from their own terminal. Privileged users can unblock any task.

Format

```
START/UNBLOCK [taskname]
```

Parameter

taskname

Names the task to be unblocked. If you do not name a task, the task initiated with the RUN command and named after the terminal is unblocked, if blocked.

Notes

See the description of STOP/BLOCK for an example.

START/UNBLOCK does not unblock tasks blocked by the Executive while waiting for system resources or for synchronization. You can only unblock tasks blocked with the STOP/BLOCK command.

The START/UNBLOCK and ABORT commands are the only means of canceling the STOP/BLOCK command.

The system automatically unblocks a blocked task when the blocked task is aborted.

INSTALL

7.8 INSTALL

INSTALL includes a specific task in the System Task Directory, thus making it known to the system.

An installed task is dormant until it is requested to run by the Executive. You can request an installed task to run through the RUN command or through a variety of Executive directives, including RQST\$ (Request), RUN\$ (Run), and SPWN\$ (Spawn).

This is a privileged command. Nonprivileged users install tasks temporarily through RUN, which includes most of the same qualifiers as INSTALL. See Section 7.2.

Format

```
INSTALL[/qualifier[s]]
File(s)? [$]filespec

INSTALL[/qualifier[s]] [$]filespec
```

Command Qualifiers

```
/[NO]CHECKPOINT
/COMMAND:"taskcommand"
/EXTENSION:n.
/[NO]INTERPRETER
/MULTIUSER_PARTITION:parname
/PARTITION:parname
/[NO]POSTMORTEM
/PRIORITY:n
/READONLY COMMON
/[NO]SLAVE
/TASK_NAME:taskname
/UIC:[g,m]
```

Parameter

[\$]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 system directory. 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 through the TASK= option of the Task Builder. See the examples.

Command Qualifiers

/[NO]CHECKPOINT

Specifies whether or not the task is to be checkpointable. The default is set at link time. This qualifier overrides the link-time checkpointability specification.

INSTALL (Cont.)

/COMMAND:"taskcommand"

Passes a command to the task you are installing. The command must be inside the quotation marks (") and not more than 40 characters long. For example, the following command:

```
$ INSTALL/COMMAND:" /LI" $PIP (RET)
```

installs PIP and then passes the /LI switch to it. This results in a directory listing.

/EXTENSION:n.

Specifies that n(decimal) additional words of address space are to be allocated to a task.

/[NO]INTERPRETER

Specifies that the task being installed is a command line interpreter (CLI). You must install a CLI using the /INTERPRETER qualifier before you can set a terminal to that CLI (SET TERMINAL/CLI:cliname) or work with the CLI by means of the MCR CLI command. The default is the /NOINTERPRETER qualifier.

/MULTIUSER_PARTITION:parname

Specifies the partition into which the read-only portion of a multiuser task is to be installed.

/PARTITION:parname

Specifies the partition into which the task is to be installed. The default is set at link time. This qualifier overrides the link-time specification.

If you name a partition that does not exist, you receive a warning message and the system attempts to install the task in the default partition GEN.

/[NO]POSTMORTEM

Specifies whether or not a Postmortem Dump (PMD) is to be generated if the task aborts because of an SST error condition. The default is set at link time. This qualifier overrides the link-time specification. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for further information.

/PRIORITY:n

Specifies the priority at which the task is to run. The default is set at link time. This qualifier overrides the default set at link-time. The argument n can be from 0 through 250(decimal).

/READONLY_COMMON

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

INSTALL (Cont.)

/[NO]SLAVE

Specifies whether or not the task is to be slaved.

Data sent to a slaved task is marked with the TI: of the sending task. When the installed task receives the data, the system sets the task's TI: to that associated with the data. The default is set at link time. This qualifier overrides the link-time specification.

Note that data may not be sent to a slaved task running on a virtual terminal.

/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. See the examples.

/UIC:[g,m]

Specifies the default UIC for the task. This task UIC determines in what protection class the task belongs and thus directly influences file access.

The brackets are required syntax.

Examples

```
$ INSTALL FATE (RET)
```

This example looks for a task image file named FATE.TSK on the default device and in the default directory and installs it as FATE in the STD, assuming no other name was specified at link time.

```
$ INSTALL $LATE (RET)
```

This example looks in the system directory (by convention, [1,54]) for a task image file named LATE.TSK and installs it as LATE in the STD, assuming no other name was specified at link time.

```
$ INSTALL BUCKING (RET)
```

This example looks for a task image file named BUCKING.TSK on the default device and in the default directory, and installs it as BUCKIN in the STD, assuming no other name was specified at link time.

```
$ INSTALL/TASK:DOOR GATE (RET)
```

This example looks on the default device in the default directory for a task image file named GATE.TSK and installs it as DOOR in the STD, overriding any task name specified at link time.

INSTALL (Cont.)

```
$ INSTALL/PARTITION:LEROY KILROY (RET)
```

This example installs the task image file named KILROY.TSK in a partition named LEROY.

```
$ INSTALL/CHECKPOINT/POSTMORTEM/PRIORITY:75/TASK:LARGO KEY (RET)
```

This example looks for a task image file named KEY.TSK in the default location and installs it in the STD under the name LARGO. The task is installed as checkpointable and at a priority of 75(decimal). If the task should exit unexpectedly, a Postmortem Dump is generated.

Notes

REMOVE counteracts INSTALL. See Section 7.9.

Tasks can also be installed through the install-run-remove form of the RUN command. Such tasks remain installed only while they are executing. While such tasks are installed, they are similar to other installed tasks. You cannot, however, count on the presence of a task installed through install-run-remove.

Several situations require tasks to be explicitly installed through INSTALL. If you wish for other tasks to be able to communicate with the task while it is not active, it must be installed. If you wish to run the task on a schedule, it must be installed and then scheduled using the RUN command. If a task is commonly used, you may wish to sacrifice pool space to improve system performance by installing the task.

You can display the attributes of installed tasks with SHOW TASKS/INSTALLED. See Section 7.12.3.2.

Error Messages

INS -- Access to common block denied

Explanation: Command attempted to access a common that was installed specifying a protection status that excludes you.

User Action: Find out why protection was established in that fashion and proceed accordingly.

INS -- Addressing extensions not supported

Explanation: Command tried to install a task with a VSECT (virtual section) in a system that does not support this feature.

User Action: None.

INS -- Base address must be on 4K boundary

Explanation: The base virtual address of the task was not on a 4K boundary.

User Action: Relink the task.

INSTALL (Cont.)

INS -- Can't install privileged task from non-privileged terminal

Explanation: A nonprivileged user attempted to install a privileged task through the install-run-remove form of RUN.

User Action: None.

INS -- Checkpoint area too small

Explanation: The area allocated for checkpointing the task was smaller than the partition into which the task was being installed.

User Action: Allocate more checkpoint space using SET DEVICE/CHECKPOINT and reenter the command.

INS -- Checkpoint space too small, using checkpoint file

Explanation: Warning message. The checkpoint space allocated in the task image file is too small for the size of the task (probably because of the /EXTENSION qualifier). The system supports dynamic allocation of checkpoint space and will therefore use a checkpoint file on the volume to contain the task if it must be checkpointed.

User Action: None necessary. The task can be relinked with more checkpoint space, or you can remove it and reinstall it with a different value for /EXTENSION if you wish.

INS -- Common block not loaded common-name

Explanation: The specified common block was linked to the task but had not been installed.

User Action: Install the common block, then install the task.

INS -- Common block parameter mismatch common-name

Explanation: Parameters of a common block did not match those in the task's label block.

User Action: Correct the mismatch.

INS -- File not contiguous

Explanation: Command attempted to install a task from a noncontiguous file. Task images must be contained in contiguous files.

User Action: Make sure you have specified the correct file. Reenter the command.

INS -- File not task image

Explanation: Command attempted to install a task from a file whose label block indicated it was not a task image.

User Action: Make sure you have specified the correct file. Reenter the command.

INSTALL (Cont.)

INS -- Illegal device ddnn:

Explanation: Warning message. The device specified at task-build time by the ASG option is not in the system. The task will be installed but may not execute as you wish.

User Action: Determine the impact of the omission and act accordingly.

INS -- Illegal device/volume

Explanation: Command specified a device that was not a valid task-residence device.

User Action: Make sure you specified the proper device and reenter command.

INS -- Illegal first APR

Explanation: A privileged task, built to run using APR 4 or 5 as its base, was too large. Using APR 4, the task was larger than 12K; using APR 5, the task was larger than 8K. If the Executive has been built to support 20K of address space, APR 5 is the only valid base APR for tasks mapping into the Executive. A privileged task that does not map into the Executive must use APR 0 as its base.

User Action: See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for a description of the /PR keyword (/PRIVILEGED qualifier to LINK).

INS -- Illegal priority

Explanation: Command specified a priority value of less than 0 or greater than 250.

User Action: Reenter command after checking proper syntax.

INS -- Illegal use of parname partition or region

Explanation: Command attempted to install task into a CPU or secondary pool partition.

User Action: Reenter command, specifying proper partition.

INS -- Length mismatch common block blockname

Explanation: The length parameter for the common block, as described in the label block for the task image, did not match the length parameter defined in the system. A task's label block data must match system data for that task before it can be installed.

User Action: See your system manager.

INSTALL (Cont.)

INS -- No checkpoint space, assuming not checkpointable

Explanation: Warning message on systems that do not support dynamic allocation of checkpoint space. Command attempted to install the task as checkpointable, but the task was not built as checkpointable and therefore had no checkpoint space allocated. Since the system does not support dynamic allocation of checkpoint space, the system considers the task to be not checkpointable.

User Action: If you wish, you can relink the task as checkpointable.

INS -- No pool space

Explanation: No dynamic memory (DSR) was available for the Common Block Directory (CBD) or Partition Control Block (PCB).

User Action: Wait. Pool space may free up soon. If it does not, see your system manager.

INS -- No room available in STD for new task

Explanation: No dynamic memory (DSR) was available for Task Control Block (TCB) in the System Task Directory (STD). Task cannot be installed.

User Action: Wait. Pool space may free up soon. If it does not, see your system manager.

INS -- Not enough APRs for task image

Explanation: The Task Builder allows you to specify the virtual base address of a task image as a multiple of 4K. Privileged tasks start at virtual 100000 (octal) to map the 16K Executive or virtual 120000 (octal) to map the 20K Executive. If the virtual base address is set too high, the task image may not be able to be mapped with the remaining mapping registers.

User Action: See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information on the /PR keyword (/PRIVILEGED qualifier to LINK).

INS -- R/O partition parname not in system defaulting to task's partition

Explanation: Warning message. Command attempted to install a read-only common in a nonexistent partition. Common was installed in task's partition.

User Action: If you wish, you can create the missing partition.

INS -- Specified partition too small

Explanation: Command attempted to install task in a partition that was smaller than the task.

User Action: Increase the size of the partition and reenter command.

INSTALL (Cont.)

INS -- Task image already installed

Explanation: The requested task image, which was checkpointable, had already been installed. On systems that do not support the dynamic allocation of checkpoint space, checkpointable tasks that have checkpoint space allocated can be installed only once. Other tasks can be installed more than once (under different names).

User Action: If the system has dynamic allocation of checkpoint spaces rebuild the task without task checkpoint space. Otherwise, make the task noncheckpointable.

INS -- Task image I/O error

Explanation: INSTALL either could not read the task image file, or could not rewrite the task image header because the device was write-locked.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your system manager.

INS -- Task image virtual address overlaps common block blockname

Explanation: The virtual addresses reserved for the task image overlap those reserved for the common block named in the message. A corrupt task image file probably caused the overlap.

User Action: Determine the state of the task image file and correct the error if you can. You may have to rebuild the task.

INS -- Task name already in use

Explanation: Command attempted to install a task with the same name as one already in the system.

User Action: Determine if the two tasks are the same. If not, install your task using the /TASK_NAME qualifier to change the name.

INS -- Too many common block requests

Explanation: Task made too many common-block requests.

User Action: Tasks are limited to seven common-block references. This error can result from an improper task build or coding errors. Find the cause of the error and correct it.

INS -- Too many LUNS

Explanation: Task requested more than 250 LUNS to be assigned.

User Action: Recheck task build and code for source of the error and correct it.

INSTALL (Cont.)

INS -- WARNING -- Privileged task overmaps the I/O page

Explanation: A privileged task that is mapped into the Executive is usually also mapped into the I/O page in order to access the KT11 registers and other devices. Many tasks do not require this access and can use the additional 4K words of virtual address space for the task image. The message warns that a task that may require access to the I/O page might have grown too large.

User Action: Determine the cause of the error and correct it if necessary.

REMOVE

7.9 REMOVE

REMOVE takes a task name out of the System Task Directory. The task is no longer installed.

REMOVE/REGION takes the name of a region out of the Common Block Directory and the partition list.

This is a privileged command.

Format

```
REMOVE
Task? taskname

REMOVE taskname
```

Parameter

taskname

Specifies the name of the task you wish to remove.

Examples

```
$ REMOVE WONTON (RET)
```

This example removes the task named WONTON from the System Task Directory. It is no longer installed.

Notes

To remove an active task, you must first abort it.

If a task is fixed, REMOVE first unfixes it and then removes it. See Section 7.10 for more information on fixing a task using the FIX command.

REMOVE cancels all time-based requests for the task to run.

REMOVE deallocates all receive-by-reference and receive-data packets for the task and detaches all attached regions. See the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual for more information.

FIX

7.10 FIX

FIX causes an installed task or region to be loaded and locked into memory. Subsequent requests for the task or region may be serviced more quickly because it is memory-resident and does not have to be loaded from the disk. This is a privileged command.

Format

```
FIX
Task? taskname

FIX[/qualifier] taskname
```

Parameter

taskname

Specifies the installed task you want to fix in memory.

Example

```
$ FIX ACNT (RET)
$
$ SHOW TASK/INSTALLED/FULL ACNT (RET)
ACNT 061724 GEN 100630 02264100-02342000 PRI - 50. DPRI - 50.
  STATUS: -EXE -CHK -PMD PRV FXD
  TI - NONE IOC - 0 EFLG - 000000 000000
$
$ UNFIX ACNT (RET)
$
$ SHO TAS INS FU ACNT (RET)
ACNT 061724 GEN 035370 0047000-00554700 PRI - 50. DPRI - 50.
  STATUS: -EXE -CHK -PMD PRV
  TI - NONE IOC - 0. EFLG - 000000 000000
$
```

In this example, the user fixed in memory the installed task ACNT. The task's status, as displayed, includes the mnemonic FXD under STATUS and also gives the task's location in the partition GEN.

Notes

UNFIX and REMOVE counteract FIX.

After a task is fixed, it need not be loaded into memory because it is already memory-resident. Requests to run the task can be serviced more quickly.

Fixed tasks remain memory-resident even after they exit or abort.

Fixing a checkpointable task makes it noncheckpointable.

Active tasks cannot be fixed. However, a task must be installed before it can be fixed.

The system can fix a task in memory only when the partition in which it is to be fixed becomes available.

FIX (Cont.)

Not all tasks will run properly when fixed. Certain tasks may require data areas to contain certain values when loaded in from the disk. The first time the task is run, these data areas may be modified and the task may run unpredictably thereafter. Tasks that initialize their data areas, and therefore do not have this problem, are called serially re-entrant.

Overlaid tasks may be fixed and, if their root segments are serially re-entrant, will run correctly. However, since their overlaid segments must still be read in from disk (unless the task was built using memory-resident overlays), not much is gained by fixing them.

Error Messages

FIX -- Partition busy

Explanation: The partition in which the task was to be fixed is occupied, so the task cannot be fixed.

User Action: Wait or install the task in another partition.

FIX -- Task already fixed

Explanation: The specified task is already fixed.

User Action: None.

FIX -- Task being fixed

Explanation: The specified task is in the process of being fixed.

User Action: None.

UNFIX

7.11 UNFIX

UNFIX frees a fixed task from memory. The effect is to allow tasks that have been waiting for space in the partition in which the fixed task resides to compete for that space. This is a privileged command.

Format

```
UNFIX  
Task? taskname
```

```
UNFIX taskname
```

Parameter

taskname

Specifies the name of the task you wish to unfix.

Example

See example in Section 7.10.

Notes

A fixed task remains fixed after it exits or aborts. Therefore, it must be explicitly unfixed or removed.

Error Message

```
UNF -- Task not fixed
```

Explanation: Command attempted to unfix a task that was not fixed.

User Action: None.

7.12 SET AND SHOW

All users can display information about partitions, active and installed tasks, the clock queue, and logical-unit-number assignments.

Privileged users can establish and eliminate partitions and change the priority of active tasks.

SET PARTITION

7.12.1 SET PARTITION

SET [NO]PARTITION creates or eliminates partitions in system memory. This is a privileged command.

A partition is a contiguous area of memory with a name, a size, a fixed base address, and a type. RSX-11M partitions can be of four types: system-controlled, task-controlled, common regions, and device commons. In addition, task-controlled partitions can have subpartitions.

Format

SET NOPARTITION:parname

SET PARTITION:parname/qualifier[s]

Base? n

Size? n

Required Qualifiers

/BASE:n

/SIZE:n

Qualifier

/TOP:arg

Partition Types

/COMMON

/DEVICE

/SYSTEM

/TASK

/TOP:arg

/[NO]SUBPARTITION:subparname

Required Argument

parname

Partition names are from one through six Radix-50 characters. Each partition (or subpartition) must have a name.

SET NOPARTITION eliminates the named partition.

Required Qualifiers

/BASE:n

Specifies the starting address of the partition. n is the physical base address specified in 64-byte units.

/SIZE:n

Specifies the size of the partition. n is the size in 64-byte units.

SET PARTITION (Cont.)

n

The argument for the base and size of the partition can be entered in a number of formats. In all formats, n is a number that is multiplied by a factor to calculate the value to be used. See Table 7-1.

Table 7-1
Base and Size Specifications for SET PARTITION

Format		Calculated Value	Base Range	Size Range
Octal	nnnn	(nnnn*100)	0<base<177777	0<size<2000
Decimal	nnnn.	(nnnn.*640)	0.<base<65535.	0<size<1024.
Octal K	nnnK	(nnn*4000)	0<base<3777K	0<size<40K
Decimal K	nnnK	(nnn.*2048.)	0<base<2047.K	0.<size<32.K

For system-controlled partitions, these values can range from 0 through 1920.K, or any valid representation of these limits. The upper limit is the size of your system's memory.

Qualifier

/TOP:arg

Moves the top boundary of the specified system according to the form the argument is stated in. The argument is a number of 64-byte blocks, expressed in one of the five following formats:

- +n Moves the top boundary of the partition up by the specified amount, making the partition larger
- n Moves the top boundary of the partition down by the specified amount, making the partition smaller
- n Makes the partition the specified size by moving the top boundary
- [+]* Moves the top boundary of the partition up as far as possible
- * Moves the top boundary of the partition down as far as possible

Partition Types

/SYSTEM

System partitions are sometimes called system-controlled partitions. They are the most common type of partition on mapped systems. All systems including DCL are mapped systems.

In system partitions, the Executive allocates available space to accommodate as many tasks as possible at one time. This allocation may involve moving resident tasks to arrange available space into a contiguous block large enough to contain a requested task.

SET PARTITION (Cont.)

/DEVICE

Device partitions are device commons, which are used by tasks to access the device registers of the I/O page. Device commons provide direct access to specialized I/O equipment, such as that used in industrial and laboratory settings. See the Processor Handbook supplied with your system for more information on the I/O page.

/TASK

Task-controlled partitions are sometimes called user-controlled partitions. Task-controlled partitions are generally used for real-time applications where explicit control of memory allocation is important. Task-controlled partitions are exclusively allocated to one task at a time unless divided into subpartitions.

/[NO]SUBPARTITION:subparname

Task-controlled partitions can be divided into as many as seven subpartitions. SET PARTITION/NOSUBPARTITION eliminates the named subpartition.

/COMMON

Common partitions are sometimes called common regions. Common partitions are used for task communications or data or code used in common by more than one task. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information on common regions.

Examples

```
$ SET PARTITION:TOMPAR/SYSTEM (RET)
Base? 002010 (RET)
Size? 1000 (RET)
$ SHOW PARTITION:TOMPAR (RET)
PAR=TOMPAR:002010:1000:SYS
```

This example creates a partition named TOMPAR with a base address of 002010 and a size of 1000(octal) bytes. The partition is system-controlled. Since this is the default, the user need not have specified /SYSTEM in the command.

The display from SHOW PARTITIONS is slightly different on RSX-11M systems in that the word MAIN appears where PAR appears in this example.

```
$ SET NOPARTITION (RET)
Partition name? TOMPAR (RET)
$ SHOW PARTITION:TOMPAR (RET)
SET -- Partition not in system
```

This example eliminates the partition TOMPAR.

```
$ SET NOPARTITION:TSTPAR/NOSUBPARTITION (RET)
Subpartition name? SMOKE (RET)
```

This example eliminates the subpartition SMOKE from the task-controlled partition TSTPAR.

SET PARTITION (Cont.)

Notes

Partitions are one of the key elements in tuning RSX-11M systems and making the best use of available memory.

You can specify the partition you wish your task to run in with the PAR option of the LINK command. You can override this specification with the /PARTITION qualifier to the INSTALL or RUN commands.

A system partition cannot include subpartitions specifically defined by the SET command. The Executive dynamically defines subpartitions in a system partition as needed for tasks installed in the partition. You should not confuse these dynamic partitions containing tasks with the task-controlled partitions of RSX-11M.

You can define subpartitions of task-controlled partitions with the SET command.

You can display the name, size, address and type of one or more partitions with the SHOW PARTITIONS command (Section 7.12.2). You can display the allocation of partitions in memory graphically with the SHOW MEMORY command (Section 8.1.7).

See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for information about tasks run in partitions.

If a partition currently contains an installed task, common, or loaded driver, you cannot eliminate it with SET NOPARTITION.

If you eliminate a task partition, you automatically eliminate any subpartitions associated with it.

Error Messages

SET -- Alignment error

Explanation: Command attempted to create a partition but the base address or size conflicted with existing partitions or with the size of physical memory.

User Action: Check command for proper values and reenter.

SET -- Nonexistent memory

Explanation: Command attempted to define a partition in nonexistent memory.

User Action: Check command for proper values and reenter.

SET -- Partition already exists

Explanation: Command attempted to define a partition with a name already in use.

User Action: Use another name.

SET PARTITION (Cont.)

SET -- Space used

Explanation: Command attempted to create a partition in a storage area already occupied.

User Action: Check command for proper values and reenter.

SET -- Task installed in partitions

Explanation: Command attempted to eliminate a partition containing an installed task.

User Action: Determine the cause of the error and proceed accordingly.

SET -- Too many subpartitions

Explanation: You attempted to define more than seven subpartitions in a task partition.

User Action: None. You are limited to seven subpartitions.

SET -- Unknown main partition

Explanation: You attempted to create subpartitions in a nonexistent task partition.

User Action: Determine the cause of the error and take appropriate action.

SHOW PARTITIONS

7.12.2 SHOW PARTITIONS

SHOW PARTITIONS displays address and content information about the partitions of the system. You can display information about all partitions or about a single partition.

Format

```
SHOW PARTITIONS[:parname]
```

Parameter

If you do not specify a partition name, all partitions in the system are displayed. The format of the display is as follows:

```
parname pcbaddr baseaddr parsize MAIN
```

If the partition includes subpartitions, this description is followed by one or more lines describing the contents of the subpartitions. The format of these lines is as follows:

```
pcbaddr baseaddr parsize partype contentname
```

In both of these displays:

parname

Specifies the name of the partition about which you want information to be displayed. When you specify a system partition, the display is as follows:

```
PAR=parname:baseaddr:parsize:SYS
```

In this display, the address and the size fields of the single partition displayed are 64-byte values (octal 100), while in the multiple partition display, byte values are displayed. See the examples.

For a task partition (user-controlled partitions), any subpartitions are also displayed as follows:

```
MAIN=parname:baseaddr:parsize:TASK
SUB=parname:subparname:baseaddr:parsize
```

pcbaddr

Specifies the address of the Partition Control Block (PCB)

baseaddr

Specifies the starting address of the partition or subpartition

parsize

Specifies the size of the partition in bytes

RUNNING TASKS

SHOW PARTITIONS (Cont.)

partype

Specifies one of the following:

MAIN	Main partition
TASK	Dynamic task subpartition - not /TASK (user-controlled)
SUB	Subpartition
DRIVER	Device driver partition
RO COM	Read-only common partition
RW COM	Read-write common partition
DEVICE	Partition is a common mapped to the I/O page
SEC POOL	Partition is a secondary pool

contentname

Indicates one of the following:

[taskname]	An inactive resident task
(TASKNAME)	An active task on RSX-11M
-dd:	Indicates specific device driver on RSX-11M
+ xxxx +	Indicates the name of the first installed task that is associated with a unnamed common, usually the read-only segment of a multiuser task
! yyyy !	Indicates the name of the common

Examples

```

$ SHOW PARTITIONS (RET)
SYSPAR 035720 00201000 00010000 MAIN
        036330 00201000 00010000 TASK  <MCR...>
        .
        .
        .
DRVPAR 035610 00224000 00140000 MAIN
        035324 00224000 00002600 DRIVER (DB:)
        035260 00227600 00001300 DRIVER (DK:)
        .
        .
        .
LDRPAR 035544 00364000 00003000 MAIN
        033774 00364000 00003000 TASK  <...LDR>
        .
        .
        .
SECPOL 035434 00427000 00050000 SEC POOL
GEN    035370 00477000 04701000 MAIN
        117210 00477000 00024000 RW COM !FCSFSL!
        041324 00523000 00004400 TASK < POOL >
        062124 00545100 00012600 TASK < LPP0 >
        .
        .
        .
        046340 00763300 00006100 TASK < TT35 >
        .
        .
        .
    
```


SHOW PARTITIONS (Cont.)

This example shows the display from SHOW PARTITIONS when no partition is named. Information on all partitions and dynamic task subpartitions in the system is displayed. DRVPAR contains a number of device drivers, each in its own subpartition. Notice that SYSPAR's base address is 00201000 and its size is 10000 (octal) bytes. See the next example.

```
$ SHOW PARTITION: SYSPAR (RET)
PAR=SYSPAR:002010:000100:SYS
```

This example shows the display from SHOW PARTITIONS when a partition is named. No information about subpartitions or about any tasks in the partition is displayed. Notice that SYSPAR's base address is given as 002010 and its size is 100 64-byte blocks.

```
$ SHOW PARTITION:TSTPAR (RET)
MAIN=TSTPAR:003657:000700:TASK
SUB-TSTPAR:SMOKE :003657:000010
```

This example shows the display when SHOW PARTITIONS names a task-controlled partition that includes a subpartition.

Notes

SHOW PARTITIONS unadorned is the equivalent of the MCR command PAR.

SHOW PARTITIONS with a partition name is the equivalent of the MCR command SET /MAIN=parname.

SHOW TASKS

7.12.3 SHOW TASKS

You can display information about active and installed tasks on the system in a number of ways. Section 7.12.3.1 describes how static information about tasks active on your terminal or elsewhere on the system can be displayed in full or brief format. Section 7.12.3.2 explains how static information about installed tasks can be displayed in full or brief format. Section 7.12.3.3 explains how static information about tasks installed from a specified device can be displayed. Section 7.12.3.4.1 explains the dynamic display of information about active tasks. Section 7.12.3.4.2 explains the dynamic display of the Active Task List. Section 7.12.3.5 explains the display of the LUNs of an installed task.

7.12.3.1 SHOW TASKS/ACTIVE

SHOW TASKS/ACTIVE displays information about active tasks in brief and full format.

Brief Format

```
SHOW TASKS/ACTIVE[:ttnn:][/qualifier[s]]
```

Command Qualifiers

```
/BRIEF  
/ALL
```

ttnn:

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.

Command Qualifier

```
/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 only.

```
/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.

SHOW TASKS (Cont.)

Examples

```
$ SHOW TASKS/ACTIVE (RET)
MCR...
SHOT5
```

This example shows the brief display from SHOW TASK/ACTIVE. In this case, the tasks are MCR..., the central command dispatcher, and SHOT5, the SHOW task itself.

```
$ SHOW TASKS/ACTIVE/ALL (RET)
LDR...
RMDEMO
SHOH2
NETACP
MCR...
SHOT5
DCLT5
DB0FCP
.
.
.
TT56
EDTT25
EDTT30
```

This example shows the brief display from SHOW TASK/ACTIVE/ALL.

Full Format

```
SHOW TASKS/ACTIVE/FULL [taskname]
SHOW TASK[:taskname]/ACTIVE/FULL
```

taskname

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.

Command Qualifier

/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 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 -psval
PC - pcval REGS 0-6 rrlval r2val r3val r4val r5val r6val
```

The elements in this display are as follows:

tcbaddr	The physical address of the Task Control Block (TCB).
parname	The name of the task's partition.

SHOW TASKS (Cont.)

pcbaddr	The physical address of the Partition Control Block (PCB).
taskaddrlimits	The base and top of the task's dynamic subpartition as physical addresses.
pri	The priority at which the task is actually running.
defpri	The default priority at which the task was built or installed to run.
STATUS	The task's status flags. These are identified in Table 7-2.
TI	The initiating terminal.
IOC	The decimal I/O count for the task.
BIO	The decimal count of I/O buffered by the terminal driver and Executive.
EFLG	Local event flags.
pswval	The Processor Status Word.
pcval	The Program Counter.
REGS	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 7-2), the PC, PSW, and other registers are not displayed.

This display shows the state of the task at the time the command was issued. You can display similar information dynamically (in real time) for a single task with the /DYNAMIC qualifier, as explained in Section 7.12.3.3.1.

Table 7-2
Task Status Flags

Status Flag	TCB Flag	Meaning
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	T2.CHK	Not checkpointable
CKD	TS.CKD	Checkpointing disabled
CKP	TS.CKP	Checkpoint in progress or checkpointed

(continued on next page)

SHOW TASKS (Cont.)

Table 7-2 (Cont.)
Task Status Flags

Status Flag	TCB Flag	Meaning
CKR	TS.CKR	Checkpoint request pending
CLI	T3.CLI	Command Line Interpreter
CMD	T3.CMD	Task is executing a CLI task
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
HLD	TS.HLD	Half-loaded, awaiting pool for task header
HLT	T2.HLT	Being terminated
MCR	T3.MCR	Activated by MCR
MSG	TS.MSG	Aborted, waiting for TKTN message
NET	T3.NET	Network protocol level
NRP	TS.NRP	Mapped to non-resident partition
NSD	T3.NSD	Cannot receive data (no send data allowed)
-PMD	T3.PMD	Suppress PMD on SST abort
OUT	TS.OUT	Out of memory
PRV	T3.PRV	Privileged
RDN	TS.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
RUN	TS.RUN	Running on another processor
SEF	T2.SEF	Stopped for event flag
SLV	T3.SLV	Slaved
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
WFERA	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 and Micro/RSX Task Builder Manual, the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual, and the RSX-11M/M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual for more information. The CDA manual has a full explanation of the contents of the TCB.

SHOW TASKS (Cont.)

Examples

```
$ SHOW TAS/ACT/FULL TT7 (RET)
TT7 066734 GEN 064244 01676500-02062100 PRI - 50. DPRI - 50.
STATUS: -CHK WFR -PMD REM MCR
TI - TT7: IOC - 1. EFLG - 000000 040000 PS-70004 PC-072132
REGS 0-6 041130 051172 000000 000040 010400 051520 001472
```

This example shows the display in full format for the task named TT7 (initiated by a RUN command from terminal TT7:). The task is running in the default partition GEN at a priority of 50. The task is not checkpointable. It is in a "wait-for" state. No PMD will be generated. The task is to be removed on exit (meaning it was installed with the install-run-remove form of RUN), and it was activated by MCR. (Many tasks requested from DCL are actually activated by MCR.)

```
$ SHO TAS/ACT/FUL (RET)
...LDR 034210 LDRPAR 033774 00364000-00367000 PRI - 248. DPRI - 248.
STATUS: STP - PMD PRV NSD FXD
TI - COO: IOC - 0. EFLG - 000001 000000 PS - 170000 PC - 120526
REGS 0-6 000162 024207 177777 066640 066610 066314 120250
RMDEMO 046320 GEN 046640 01367700-01415100 PRI - 225. DPRI - 225.
STATUS: -CHK WFR -PMD PRV MCR
TI - TT62: IOC - 0. EFLG - 000035 040000 PS - 170010 PC - 125014
REGS 0-6 000002 131316 131436 000000 135055 000000 121240
MCR... 116464 SYSPAR 036330 00201000-00211000 PRI - 160. DPRI - 160.
STATUS: STP -PMD PRV CLI NSD CAL
TI - TT0: IOC - 0. EFLG - 000001 040000 PS - 170000 PC - 124412
REGS 0-6 000000 120556 000400 120512 000100 120572 120442
DCL... 115530 GEN 037344 00624700-00634700 PRI - 160. DPRI - 160.
STATUS: STP -PMD PRV CLI NSD CAL
TI - TT10: IOC - 0. EFLG - 100001 040000 PS - 170000 PC - 124470
REGS 0-6 025550 025550 000000 120426 123530 000000 120362
SHOT10 066314 GEN 066610 02221700-02261700 PRI - 160. DPRI - 160.
STATUS: -PMD REM PRV
TI - TT10: IOC - 0. EFLG - 000001 040000 PS - 170017 PC - 121202
REGS 0-6 000000 131574 050712 000000 000000 000000 001216
DBOOF1 117570 GEN 036770 00537000-00624700 PRI - 149. DPRI - 149.
STATUS: STP ACP -PMD PRV NDS CAL ROV
TI - COO: IOC - 0. EFLG - 000002 040001 PS - 170000 PC - 120546
:
:
LPP0 061444 GEN 062504 00725200-00740000 PRI - 80. DPRI - 80.
STATUS: STP PRV
TI - COO: IOC - 0. EFLG - 000400 140000 PS - 170000 PC - 120644
REGS 0-6 121526 121640 000115 124676 000000 000001 120342
PARENT TASK NAME - QMG...
LPP1 062024 GEN 062374 01351200-01364000 PRI - 80. DPRI - 80.
STATUS: STP PRV
TI - COO: IOC - 0. EFLG - 000400 040000 PS - 170000 PC - 120644
REGS 0-6 120362 000061 000060 140164 140440 000000 120342
PARENT TASK NAME - QMG...
AT.T10 041174 GEN 040640 02062100-02142000 PRI - 80. DPRI - 80.
STATUS: SEF STP WFR -PMD REM PRV MCR
TI - TT10: IOC - 0. EFLG - 000000 140000 PS - 170010 PC - 141604
REGS 0-6 123551 123416 000122 115530 000000 000000 120664
```

SHOW TASKS (Cont.)

```

QMG... 061340 GEN      056354 01327700-01351200  PRI - 75.  DPRI - 75.
STATUS:  STP PRV
TI - COO:  IOC - 0.  EFLG - 000000 040000  PS - 170000  PC - 120656
REGS 0-6 121750 123350 122350 001006 000000 000000 120372
EDIT10 057420 GEN      057340 02142000-02262100  PRI - 65.  DPRI - 65.
STATUS:  WFR -PMD REM
TI - TT10: IOC - 0.  EFLG - 000002 100000  PS - 170000  PC - 006706
REGS 0-6 000001 000000 017764 000000 103666 1015708 001700
PARENT TASK NAME - AT.T10

```

⋮

This example shows the display from SHOW TASKS/ACTIVE/FULL. Tasks are displayed in order of priority. Note the tasks displayed below the first break. Tasks LPP0 and LPP1 are line-printer processors. They are spawned by the Queue Manager. AT.T10 represents an indirect command file that spawns EDT by including an EDIT/EDT command, but the resulting task EDIT10 is not slaved.

7.12.3.2 SHOW TASKS/INSTALLED

SHOW TASKS/INSTALLED displays information about installed tasks in either brief or full format.

Format

```
SHOW TASKS[:taskname]/INSTALLED[/qualifier]
```

Command Qualifiers

```
/BRIEF
/FULL
```

Parameter

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.

Command 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 ddn:-lbn [memstate]
```

The elements in this display are as follows:

```
taskname      The name of the task
ident         The task version identification (or the version of
              the prototype task)
```

SHOW TASKS (Cont.)

parname The partition in which the task is installed
 priority The task's priority
 size The size of the task in bytes
 ddnn: The device from which it is to be loaded
 lbn The logical block number of its disk address
 memstate 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 Section 7.12.3.1 for a description of the display.

Examples

```
$ SHOW TASKS/INSTALLED (RET)
LDR... 12.15 LDRPAR 248. 00002500 LB0:-00163310 FIXED
TKTN   04.7  TKNPAR 248. 00010000 LB0:-00171022
RMDEMO V1.03  GEN    200. 00026700 LB0:-00171022
MTAACP 0013  GEN    200. 00013600 EM0:-00000315
FILMSG V0010  GEN    200. 00025500 LB0:-00167207
NETACP V02.00 GEN    200. 00025500 DB0:-00644640
EVP...  V01.00 GEN    199. 00013100 DB0:-00644640 CHECKPOINTED
...MCR  1.1   GEN    160. 00020300 LB0:-00162467
SHOT5  1.1   GEN    160. 00020300 LB0:-00162467
...MOU  2502  GEN    160. 00037700 LB0:-00222416
...DCL  00    GEN    160. 00035600 DB0:-00212612
MCR...  2.02  SYSPAR 160. 00010000 LB0:-00162443
.
.
.
```

This example shows the brief format of the display from SHOW TASKS/INSTALLED. LDR..., the loader, is fixed in memory. The task EVP... is checkpointed. The final entry shown, MCR..., is the system command dispatcher. See the next example.

SHOW TASKS (Cont.)

```

$ SHOW/TASK:MCR.../INSTALLED/FULL (RET)
MCR... 110160 SYSPAR 11454 00202700-00212700 PRI - 160 . DPRI - 160
STATUS: STP -PMD PRV MCR CLI NSD CAL
TI - TT5: IOC - 0. BIO - 0. EFLG - 000001 040000 PS - 170000
PC - 122436 REGS 0-6 000000 120476 000102 120516 057020 040220 120366
    
```

This example shows the full format of the display from SHOW TASKS/INSTALLED. In this case, the user named a single task. The default is to show the full display for all installed tasks. The status flags in the second line of the display are explained in Table 7-2.

Notes

The brief format of SHOW TASKS/INSTALLED is the MCR TAS command. The full format of SHOW TASKS/INSTALLED is the MCR TAL command.

The /DYNAMIC qualifier enables you to watch the full format display as it is updated. See Section 7.12.3.4.

7.12.3.3 SHOW TASKS/INSTALLED/DEVICE

SHOW TASKS/INSTALLED/DEVICE displays the names and status of all tasks installed from a specified device.

Format

```
SHOW TASKS/INSTALLED/DEVICE:ddnn:
```

Example

```

$ SHOW TASKS/INSTALLED/DEVICE:DR0: (RET)
DR00F1 M0323 SYSPAR 149. 00055700 LB0:-00475656 FIXED
F11ACP M0323 GEN 149. 00044000 LB0:-00315566
DB01F1 M0323 GEN 149. 00044000 LB0:-00315566
DB07F1 M0323 SYSPAR 149. 00044000 LB0:-00315566 FIXED
ERRLOG V1.01 GEN 148. 00042200 LB0:-00335276
PMD... 04.0 GEN 140. 00015200 LB0:-00503535
HRC... 02 GEN 140. 00055100 LB0:-00505331
SYSLOG 1.02 GEN 130. 00016300 LB0:-00504250
...INS 06 GEN 100. 00035400 LB0:-00502750
    
```

This display contains the following information, from left to right:

- Task name
- Task version identification
- Partition name
- Task priority
- Task size
- Load device identification
- Logical block number on the load device
- Task memory state

SHOW TASKS (Cont.)

On a processor with less than 124K words of memory, the task size field (fifth column) contains 6-digit rather than 8-digit numbers.

FIXED indicates that the task is fixed in memory. CHECKPOINTED indicates that the task has been swapped out of memory to make room for a higher priority task.

If the task version identification is missing (the rest of the line is offset to the left) or is irrelevant, the task was installed from a disk that has since been removed from the drive or replaced by another one.

7.12.3.4 SHOW TASKS/DYNAMIC

SHOW TASKS/DYNAMIC displays a moving picture on a video terminal of either a single task's activity, or of all or part of the Active Task List.

The command also works on hard-copy terminals, providing a snapshot display that is different in format from the conventional SHOW TASKS display.

This command is privileged if the Resource Monitoring Display (RMD) is not installed on the system. The command is nonprivileged if RMD is installed under the name ...RMD, or if your system supports flying installs. See your system manager for information.

7.12.3.4.1 SHOW TASK:taskname/DYNAMIC

SHOW TASK:taskname/DYNAMIC displays a moving picture of the task's activity on a video terminal or a snapshot display on a hard-copy terminal.

Format

SHOW TASK:taskname/DYNAMIC[/qualifier]

Command Qualifier

/RATE:n

Argument

taskname

Specifies the task you wish to inspect.

Qualifiers

/DYNAMIC

Specifies that you wish the moving display. This function is performed by the RMD task. RMD has four display pages:

1. Task, which displays a task header
2. Active, which displays all or part of the Active Task List

SHOW TASKS (Cont.)

3. Memory, which displays the contents of memory
4. Help, which provides help on RMD

Once you have invoked any one of these pages, through either a SHOW TASK/DYNAMIC command or a SHOW MEMORY command, you can move from one page to the other without leaving RMD. The first character of each page name is a command to RMD to go to that page. In addition, an ESC entered from a page permits you to enter setup commands for that page. See Section 8.1.7 for more information on the features of RMD.

The setup commands for the Task page permit you to change the task header being displayed. The task you name remains the default display for the Task page. This means you can observe a single task header, then jump to the Memory page or the Active page, and then back to the Task page and the named task will still be on display.

/RATE:n

Allows you to set the rate at which the RMD display screen is to be replotted on the first display. n is the number of seconds between replots. The default n is 1. You can change the rate by pressing ESC and entering a new rate. The rate setting returns to 1 as soon as you change pages. If you wish a slower rate, you must reset the rate each time you change the page.

Use this qualifier to slow down the display.

Task headers may change more often than once per second, but once per second is the most rapid rate available.

Example

```
$ SHOW TASK:CORT5/DYNAMIC (RET)
```

The screen is cleared and the following display appears:

```
RSX-11M V4.2 BL38 512K                               14-APR-85 18:15:33

Task: CORT5 Partition: GEN STATUS: -CHK -PMD REM PRV CLI
Own: TT5: I/O = 0. DPri: 160 Pri: 160 SPri: 155 Len: 000356000

R0 = 022120 R1 = 000764 R2 = 002230 R3 = 000746 R4 = 024626 R5 = 022136
PC = 005330 PS = 170000 SP = 022224 $DSW = 0. Eflg = 0000001 040000

LUN File                                             LUN File
---- ----
1. SY0:
2. DB0: [3,33]DEEP.DAT
3. SY0:
4. SY0:
5. TI0:
6. CL0:
7. DB0:
```

SHOW TASKS (Cont.)

This display changes once per second and shows the task header for a task named CORT5. The top line of the display identifies the operating system by name, version, base level and memory size, and also gives the date and time. The next line of the display includes the task name, the partition in which the task is running, and the task's status flags. (See Table 7-2 for an explanation of the status flags.) The next line lists the task's owner (terminal, not UIC); the outstanding I/O count; the default priority, running priority, and swapping priority; and the length of the task in octal bytes (top minus base). The next two lines display the contents of the task's general-purpose registers, program counter, Processor Status Word, stack pointer, Directive Status Word, and local event flags. The remainder of the display shows the logical-unit-number assignments of as many as 25 LUNs. (Tasks can have as many as 250 LUNs.) This task has conventional LUN assignments except for LUN 2, which is an open file.

Notes

This display is a useful tool for understanding how tasks run on the system. However, you must have a thorough knowledge of how tasks run in the RSX-11M environment to understand all the elements of this display.

Tasks built including ODT can be observed using this command, but if you are single-stepping through the task or otherwise using breakpoints, the registers displayed will be those of the ODT task, not those of the named task.

This display is particularly useful for observing complex assemblies, as you can tell how far the assembly has gone by which files are open. You may also be interested in observing the Task Builder at work.

7.12.3.4.2 SHOW TASKS/ACTIVE/DYNAMIC

SHOW TASKS/ACTIVE/DYNAMIC displays a moving picture of the Active Task List on a video terminal or a snapshot display on a hardcopy terminal.

Format

```
SHOW TASKS/ACTIVE/DYNAMIC[/qualifier[s]]
```

Command Qualifiers

```
/OWNER:ddnn:  
/PRIORITY:n  
/RATE:n
```

Command Qualifiers

```
/OWNER:arg  
ddnn:  
ALL
```

Specifies the terminal or other device from which the tasks displayed are being run. You can name any device, pseudo device, logical device name, or terminal number for this argument. The default argument is ALL, which displays tasks owned by all devices on the system.

RUNNING TASKS

SHOW TASKS (Cont.)

You can change the owner displayed by pressing the ESC key and entering a new device name.

Note that this qualifier differs from other "owner" qualifiers in DCL in that it names a terminal and not a UIC as the owner.

/PRIORITY:n

Specifies the highest priority to be displayed. The default is 250, the highest priority on the system.

Use this qualifier to shorten the display to within one screen's worth.

You can change this value by pressing the ESC key and entering a new priority.

/RATE:n

Allows you to set the rate at which the RMD display screen is to be replotted on the first display. The argument n is the number of seconds between replots. The default value for n is 1. You can change the rate once the display begins by pressing the ESC key and entering a new rate. The rate setting returns to 1 as soon as you change pages. If you wish a slower rate, you must reset the rate each time you change the page.

Use this qualifier to slow down the display.

The Active Task List may change more often than once per second, but once per second is the most rapid rate available.

Example

```
$ SHOW TASKS/ACTIVE/DYNAMIC (RET)
```

The screen is cleared and the following display appears:

```
RSX-11M V4.2 BL38    512K          OWN=ALL          14-APR-85 19:08:55

  Name  Length  TI      Pri      I/O Status flags
LDR... 00002500 CO0    248.    0. -CHK FXD STP -PMD PRV NSD
RMDEMO 00026700 TT0    225.    0. -CHK WFR -PMD PRV MCR
SHOH0  00026700 HT0    225.    0. -CHK -PMD PRV MCR
NETACP 00025500 CO0    200.    0. STP ACP -PMD PRV NSD
EVP... 00014200 CO0    199.    0. STP -PMD PRV
DCLT5  00035600 TT5    160.    0. -CHK STP REM PRV CLI
DB0FCP 00044000 CO0    149.    0. -CHK STP ACP -PMD PRV NSD
      .
      .
LP0     00014100 CO0    70.     0. STP PRV SLV
TT5     00002600 TT5    50.     1. -CHK WFR -PMD REM MCR
```

SHOW TASKS (Cont.)

The screen changes once per second and displays the Active Task List. The first line identifies the operating system, version number, base level, memory size, the current owner of the tasks being displayed, and the date and time. Since the current owner is ALL, the remainder of the display is the Active Task List itself. The display includes the task name, its length in octal bytes, its running priority, outstanding I/O count, and the status flags for each task. (See Table 7-2 for an explanation of the status flags.)

```
$ SHOW TASKS/ACTIVE/DYNAMIC/OWNER:TT5: (RET)
```

The screen is cleared and the following display appears:

```
RSX-11M V4.2 BL38 512K          OWN=TT5:          21-APR-85 14:57:19
Name   Length   TI    Pri I/O Status flags
SHOT5  00026700   TT5  0.  CHK -PMD  REM  PRV  MCR
```

This example displays the Active Task List for a single terminal. The only task active at the terminal is the SHOW command itself.

Notes

The RMD display is in order of priority. When a new task comes in, all tasks below it in priority are displaced on the screen.

See Section 8.1.7 for more information on RMD.

7.12.3.5 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
```

Argument

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.

Tasks installed through the install-run-remove form of RUN do not have any static LUN assignments.

SHOW TASKS (Cont.)

Examples

```
$ SHOW TASK:TEWKSY/LOGICAL_UNITS (RET)
DB1: 1.
DB1: 2.
DB1: 3.
DB1: 4.
TI0: 5.
CLO: 6.
```

This example shows the display.

```
$ SHOW TASK:TT7/LOGICAL (RET)
SHO -- Task not in system
```

This example shows the error message received when you attempt to show the LUNs of a task installed through the install-run-remove form of RUN.

Notes

You can display information about a terminal's logical assignments with SHOW ASSIGNMENTS, Section 5.11.1.

You can display information about the LUNs of a task installed with the install-run-remove form of the RUN command with the SHOW TASKS/DYNAMIC command. See Section 7.12.3.4.

SHOW TASK/LOGICAL_UNITS is the equivalent of the MCR LUN command.

For more information on logical units, see Chapter 5. See also the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.

Tasks can dynamically alter their LUN assignments through the ALUN\$ Executive directive.

Error Messages

SHO -- No LUNS

Explanation: Informational message. The task did not have any logical units.

User Action: None.

SHO -- Task not in system

Explanation: Informational message. Command specified a task that is not in the system, or that was installed by means of the install-run-remove form of RUN.

User Action: SHOW ASSIGNMENTS displays logical device assignments for your terminal. SHOW TASKS/DYNAMIC can display LUN assignments for tasks installed with the install-run-remove form of RUN.

SET PRIORITY

7.12.4 SET PRIORITY

SET PRIORITY alters the priority of an active task. This is a privileged command.

Format

```
SET PRIORITY
Priority? pnum
Task? taskname
SET PRIORITY:n taskname
```

Parameters

pnum

Specifies the new priority you wish to assign to the task. Priority numbers are in the range of from 1 through 250. n is octal or decimal. You must add a decimal point to decimal numbers.

taskname

Specifies the name of the active task whose priority you wish to alter.

Example

```
$ SHOW TASK:TT7/FULL/ACTIVE (RET)
TT7 066734 GEN 064244 01676500-02062100 PRI - 50. DPRI - 50.
STATUS: -CHK WFR -PMD REM MCR
TI - TT7: IOC - 1. EFLG - 000000 040000 PS-70004 PC-072132
REGS 0-6 041130 051172 000000 000040 010400 051520 001472
$ SET PRIORITY (RET)
TASK? TT7 (RET)
PRIORITY? 199 (RET)
SET -- Syntax error
$
$ SET PRIORITY TT7 199. (RET)
$ SHOW TAS ACT FUL TT7 (RET)
TT7 066734 GEN 064244 01676500-02062100 PRI - 199. DPRI - 199.
STATUS: -CHK WFR -PMD REM MCR
TI - TT7: IOC - 1. EFLG - 000000 040000 PS-70004 PC-072132
REGS 0-6 041130 051172 000000 000040 010400 051520 001472
$
$ SET PRIORITY TT7 225 (RET)
$ SHOW TAS ACT FULL TT7 (RET)
TT7 066734 GEN 064244 01676500-02062100 PRI - 149. DPRI - 149.
STATUS: -CHK WFR -PMD REM MCR
TI - TT7: IOC - 1. EFLG - 000000 040000 PS-70004 PC-072132
REGS 0-6 041130 051172 000000 000040 010400 051520 001472
```

This example shows the effects of SET PRIORITY. The user first displayed the priority of task TT7 with SHOW TASKS/FULL/ACTIVE. The user then issued SET PRIORITY to change the task's priority to 199 but did not terminate the priority number with a decimal point. The system rejected the command because it recognized that a number containing a 9 was not an octal number. The user then issued the command with the decimal point and it was

SET PRIORITY (Cont.)

processed. The user then attempted to change the task's priority to 255, again omitting the decimal point. This time the command was processed without an error message, but the priority was changed to 255(octal), which is 149(decimal).

Notes

SET PRIORITY changes both the running and default priority of a task. A task can change its own priority through the ALTP\$ Executive directive. With this directive, only the running priority is changed.

There is no SHOW PRIORITY command. You must use SHOW TASKS/ACTIVE/FULL instead.

SET PRIORITY is the equivalent of the MCR ALT command.

SHOW CLOCK_QUEUE

7.12.5 SHOW /CLOCK_QUEUE

SHOW /CLOCK_QUEUE displays information about tasks currently in the clock queue. The information consists of the task names, the next time each task is to be run, and each task's reschedule interval, if one was specified.

Format

```
SHOW /CLOCK_QUEUE
```

Example

```
$ SHOW /CLOCK_QUEUE (RET)
LIONEL Scheduled at 29-MAY-85 17:41:31:51 Reschedule interval 05 Secs
ONEUP Scheduled at 29-MAY-85 17:48:44:05 Reschedule interval 20 Mins
CALDOR Scheduled at 29-MAY-85 18:00:00:00 Reschedule interval 02 Hours
TEWKSY Scheduled at 30-MAY-85 10:00:00:00
```

This example shows the display from SHOW /CLOCK_QUEUE. Installed tasks are placed in the clock queue through the scheduling options of the RUN command. See Section 7.2.2.

The task LIONEL is scheduled to run at 17:41:31:51 (hours, minutes, seconds, ticks) and every 5 seconds thereafter.

The task ONEUP is scheduled to run at 17:48:44:05.

The task CALDOR is scheduled to run at 18:00:00:00 and every two hours thereafter.

The task TEWKSY is scheduled to run at 10:00:00:00 on the next day.

Notes

SHOW /CLOCK_QUEUE is the equivalent of the MCR CLQ command.

Installed tasks are placed in the clock queue through the privileged form of the RUN command. See Section 7.2.2.

This command has nothing to do with queues under the control of the Queue Manager. Entries in Queue Manager queues are displayed through SHOW QUEUE. See the RSX-11M-PLUS Batch and Queue Operations Manual.

CHAPTER 8

SYSTEM CONTROL

The commands in this chapter are used to control and display certain system level functions. The SHOW commands in this chapter are nonprivileged, but all other commands in the chapter are privileged. Other commands needed for system control may be found in other chapters of this manual, in the RSX-11M/M-PLUS MCR Operations Manual, and in the RSX-11M/M-PLUS System Management Guide.

8.1 SET AND SHOW

You can set and display the time and date, the system directory, and the size of the dynamic storage region (pool). You can display the logged-in users and the current use of memory on the system, and you can enable and disable logins to the system.

8.1.1 SET DAYTIME

SET [DAY]TIME sets the current date and time.

Format

```
SET [DAY]TIME [date] [time]
```

Parameters

Either the time or the date, or both, can be specified in the command. They can be specified in either order. If you do not specify a time or a date, you will be prompted. The date field and the time field must be separated by a space.

date

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.

SET [DAY]TIME

time

Specifies the time. Time is entered in the following format:

hh:mm[:ss] Where hh is the number of the hour in the
24-hour format and mm is the number of the
minute. ss defaults to 00

Examples

```
$ SET TIME (RET)
Time? 8:05 (RET)
```

This example sets the time at 8:05 on the current day.

```
$ SET DAY (RET)
Date? 16-JUN-85 (RET)
```

This example sets the date at June 16, 1985. The time setting is unchanged.

```
$ SET DAY (RET)
Date? 06/14/85 17:14 (RET)
```

This example sets the date at June 14, 1985, and the time at 17:14:00. The date will not be displayed in the format entered but as 14-JUN-85.

```
$ SET DAYTIME 06-JUN-44 04:00:00 (RET)
```

```
$ SET TIME 04:00:00 06-JUN-44 (RET)
```

The examples are equivalent. Both set the date at June 6, 1944, and the time at 04:00 hours.

Notes

Regardless of the prompt, you can enter the date or time in either order.

SHOW [DAY]TIME SET SYSTEM

8.1.2 SHOW [DAY]TIME

SHOW [DAY]TIME displays the current time and date. The time is in 24-hour format and the date is formatted as dd-mmm-yy.

Format

```
SHOW [DAY]TIME
SHOW TIME
SHOW DAY
```

Example

```
$ SHOW TIME (RET)
04:00:22 06-JUN-44
```

8.1.3 SET SYSTEM

SET SYSTEM establishes certain characteristics of the system. It is a privileged command.

Format

```
SET SYSTEM[/qualifier]
```

Command Qualifiers

```
/DIRECTORY:[g,m]
/EXTENSION_LIMIT:n
/[NO]LOGINS
/NETWORK_UIC:[g,m]
/PACKETS:n
/POOL:top:max:total
/POOL/LIMITS:arg
    HIGH=n
    LOW=n
    MINIMUM_SIZE=n
    TASK_PRIORITY=n
```

Command Qualifiers

```
/DIRECTORY:[g,m]
```

Establishes the identity of the system directory. The directory you name must already exist. This command does not create a directory.

The /DIRECTORY qualifier is the default for SET SYSTEM.

See the discussion in Section 8.1.3 for information on the use of this command on RSX-11M systems.

```
/EXTENSION_LIMIT:n
```

Establishes the maximum size to which a task can extend itself by means of the EXTEND TASK directive. n can be entered in all the formats used in the SET PARTITION command. See Section 7.12.1.

SET SYSTEM (Cont.)

See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual and the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual for more information on the Extend Task directive.

/[NO]LOGINS

Enables or disables logins.

If you issue SET SYSTEM/NOLOGINS and then log out, you will have to reboot the system to log in again unless there is another privileged user logged in.

/NETWORK_UIC: [g,m]

Specifies the directory ([g,m]) in which all DECnet-related tasks are stored. The qualifier applies only to systems that selected the external communication products option during system generation.

/PACKETS:n

Specifies the maximum number of I/O packets to be retained in a separate queue for use by the QIO directive. Changing the number of retained packets helps optimize the servicing of QIO requests. n can be from 0 through 15. The default is 5.

This value is a system-generation option on RSX-11M systems.

This command is equivalent to the SET /MAXPKT command in MCR and VMR.

SHOW SYSTEM/PACKETS displays this limit and also the number of packets currently available.

/POOL:top:max:total

Increases the size of the system pool (the dynamic storage region). The top argument is the number of 32-word blocks (decimal). The max argument is the maximum size in decimal words. The total argument is the total size in decimal words.

/POOL/LIMITS:arg[s]

HIGH=n
LOW=n
MINIMUM_SIZE=n
TASK_PRIORITY=n

Sets the pool limit parameters used by the Pool Monitor Task (PMT). The following list defines the parameters:

HIGH The high-pool limit in bytes. The default is 1600(decimal) bytes.

LOW The low-pool limit in bytes. The default is 600(decimal) bytes.

MINIMUM_SIZE The minimum byte size of the largest free pool block required for avoiding low-pool action by the Pool Monitor Task. The default is 200(decimal) bytes.

SET SYSTEM (Cont.)

TASK_PRIORITY The base task priority, which is the lowest priority a nonprivileged task can have and still be eligible for memory contention during times of low pool. This priority remains in effect until pool conditions improve. Note that only those tasks requested while the system is in a low-pool state are affected. The default is 51(decimal) bytes.

The following list defines the limits of the parameters:

```
84. <= low <= high <= total size of system pool
84. <= minimum <= high
0. <= priority <= 250.
```

See the RSX-11M/M-PLUS System Management Guide for information on determining the values of these parameters.

Examples

```
$ SET SYSTEM (RET)
System UFD? [3,54] (RET)
$
```

```
$ SET SYSTEM [3,54] (RET)
```

These commands are equivalent. They establish the system directory as UFD [3,54] on pseudo device LB:.

```
$ SET SYSTEM/EXTENSION_LIMIT:1200 (RET)
```

This command sets the maximum size to which a task can extend itself through the Extend Task directive to 1200 (octal) 32-word blocks (decimal) or 30K words.

SHOW SYSTEM

8.1.4 SHOW SYSTEM

SHOW SYSTEM displays information about the current system.

Format

SHOW SYSTEM[/qualifier]

Command Qualifiers

/CLI
 /DIRECTORY
 /EXTENSION LIMIT
 /NETWORK_UIC
 /PACKETS
 /POOL
 /POOL/LIMITS

Command Qualifiers

/CLI

Displays information about each command line interpreter (CLI) on the system. The information includes the name of the CLI, the name of the task that implements the CLI, the number of terminals set to the CLI, and several status flags for the CLI.

The status flags are as follows:

ACT	CLI is active
DSB	CLI is disabled
PRV	CLI is available to privileged users only
RST	CLI is restricted

For more information about adding CLIs to the system, see the RSX-11M/M-PLUS MCR Operations Manual and the RSX-11M/M-PLUS System Management Guide.

There is no SET SYSTEM/CLI command in DCL.

/DIRECTORY

Displays the current system directory. The display specifies SYSUIC, but the system directory rarely has a UIC associated with it; it is a directory. See Section 8.1.5 for more information.

/EXTENSION_LIMIT

Displays the maximum size to which a task can extend itself by means of the Extend Task system directive. The size is displayed as an octal number of 32-word blocks (decimal).

/NETWORK_UIC

Displays the current network directory.

SHOW SYSTEM (Cont.)**/PACKETS**

Displays the maximum number and the current number of available I/O packets in the format:

MAXPKT=m.:c.

where m is the maximum number and c is the current number of available I/O packets.

Generally, some number of 18-word I/O packets are set aside in pool and retained in a separate queue for use by the QIO directive. This number of packets (referred to as MAXPKT) is used to optimize the servicing of QIO requests.

/POOL

Displays the current status of the system dynamic pool as three numbers, flor: lfb: tps, where flor is the first location available for user partitions, lfb is the largest free block in decimal words, and tps is the total pool space in decimal words.

The SHOW MEMORY command (Section 8.1.7) displays considerably more information about the pool.

/POOL/LIMITS

Displays the current settings for the pool limit parameters. The display is in the following format:

PLCTL=high:low:minimum:priority

The elements are as follows:

high	The high pool limit in bytes
low	The low pool limit in bytes
minimum	The minimum byte size of the largest free pool block
priority	The lowest priority a nonprivileged task can have to compete for memory during periods of low pool

8.1.5 SYSTEM DIRECTORIES

On RSX-11M systems, the system image file and all system tasks and utilities are kept in the system directory on pseudo device LB:. You can display the current setting for this directory with the SHOW SYSTEM/DIRECTORY command. Privileged users can change the directory with the SET SYSTEM/DIRECTORY command.

When you issue a RUN or INSTALL command and precede the file specification (that is, the task image file name) with a dollar sign (\$), you are instructing the system that the task you wish to run or install is in this directory.

SHOW USERS

8.1.6 SHOW USERS

SHOW USERS on RSX-11M systems displays the terminal number and login UIC for all logged-in users.

Format

```
SHOW USERS
```

Examples

```
$ SHOW USERS (RET)
TT0:   [3,54]
TT2:   [7,40]
TT5:   [301,370]
TT7:   [200.1]
TT16:  [7,42]
TT51:  [7,26]
TT54:  [303,5]
HT2:   [7,43]
```

This example shows the SHOW USERS display for an RSX-11M system. The first column identifies the terminal. The second column is the login UIC for nonprivileged users and the current UIC for nonprivileged users. HT2: is a DECnet host terminal.

SHOW MEMORY

8.1.7 SHOW MEMORY

SHOW MEMORY invokes the Resource Monitoring Display (RMD), a system utility program. This program displays in a graphic manner the status of much of the system. It is, in effect, a display of the contents of the system pool (dynamic storage region).

The RMD program is useful for monitoring the general activity of the system. It is also useful viewing for a new user to see how the operating system operates. However, you should understand that the graphic display is approximate and cannot be used for critical measurement.

Format

SHOW MEMORY

Example

\$ SHOW MEMORY **(RET)**

The following example shows the SHOW MEMORY display for RSX-11M. The circled numbers are keyed to the explanation.

```

      ①                ②                ③                ④
RSX-11M V4.2 BL38      (KERMIT) 512K                12-MAY-85 10:59:16
⑤ TASK= *IDLE*        ⑥ FREE= SY0:18470. DB2:14542.
                        DB1:62322. DB3:DMO                PARS
⑦ POOL=1686.:3168.:55.
  1686.:3168.:55.
IN:  TDMDD F    LPRQ . D D    . . R F    E    .
14  TICBD C    POMM . B B    . . M l    D    .
141K :RR:: S    LODG . 0 0    . . D l    T    .
⑨ OUT: C.    R    LE. A l F    E C T A    T    L    ⑩
4    O.    E    M. T F C    D A 5 C    l    B
31K  M.    S    O. . l P    T . 4 P    7    Z
      (!>)) !-! <>>>-->-->-->-->--><->--> <----->----->
0*****32*****64*****96*****128*****160*****192*****224*****
E-P-CT-TD-C-C-T---D-----
-----
256*****288*****320*****352*****384*****416*****448*****480*****
ERRSEQ
0.
      ⑪                ⑫

```

- ① Operating system type, version number, and base level.
- ② Six-character name that is a DECnet node name (if DECnet is running on your system) or the system name that you selected in SYSGEN (if DECnet is not running on your system).
- ③ Size in K words of the system memory.
- ④ Current date and time.
- ⑤ Name of the task that is currently executing or, if none is executing, *IDLE* (Executive executing the idle loop).

SHOW MEMORY (Cont.)

- ⑥ Number of free blocks on four Files-11 devices in your system. If a device is dismounted, RMD displays "DMO." If a device is off line, RMD displays "OFL."

- ⑦ Pool (dynamic storage region) information in the format:

POOL=X:Y:Z

X Number of words in the largest free block in pool

Y Number of free words in pool

Z Number of free words in pool

The second line records the worst case of pool since you invoked RMD. This line is most useful if RMD has been running on a slaved terminal since the system was last bootstrapped.

- ⑧ Partitions in the system using the format:

partitionname:type

type

For RSX-11M systems, one of the following:

T Task partition
 S Subpartition
 D System-controlled (dynamic) partition
 C Command partition

- ⑨ Number of tasks in memory and amount of memory they use and number of active tasks swapped or checkpointed out of memory and the amount of memory they would require.

- ⑩ Name of each task, common, or driver in memory and its location in memory using the following symbols to designate size, type (task, common, or driver), and other attributes:

Symbol	Attribute
< >	Active task
[]	Task not active but occupies memory
!!	Named common
++	Unnamed common (displayed name is first attached task)
()	Loaded driver using device mnemonic
----	Task not fixed in memory
====	Task fixed in memory

The hyphens and equal signs represent the amount of memory that each task, driver, or common occupies. Where the display shows only one delimiter and no hyphens or equal signs, the open delimiter is in the same location as the closing delimiter of the preceding task.

SHOW MEMORY (Cont.)

- ⑪ Partition size and location. The beginning of each partition is marked with the same symbols as those listed previously plus the following additions for both operating systems:

```

E   Executive
P   Pool

```

The lines of asterisks are proportional representations of the amount of memory occupied by each partition. The numbers are in 1K-word increments. Each numerical character also represents the same amount of memory as an asterisk.

- ⑫ System error count sequence recorded by the Error Logger (always zero if Error Logger not present in your system).

Notes

RMD must be installed under the name ...RMD for this command to work.

RMD adjusts its display to your terminal model. If you are at a video terminal, the display changes as you watch it. If you are at a hardcopy terminal, a snapshot of the display as of the moment you issued the command is printed on your terminal.

Often, RMD is set running on a slaved terminal near the computer as part of the system bootstrap procedure.

Task locations and sizes displayed by RMD should be considered approximate. RMD is primarily designed to provide a visual display and system debugging tool. It does not measure task size or location accurately.

In general, RMD is designed to allow either an operator or user to view the current state of the operating system. The information is presented in a graphic, easily comprehensible form.

There are a number of specific instances in which RMD can help to locate certain system lockout problems or bugs in application and/or system-level software:

- Bugs and other situations that cause exhaustion of system dynamic memory (resulting in unexplained system hanging) may be detected and more easily reproduced and diagnosed.
- Task checkpointing and swapping can be monitored and, in systems with limited memory, deadlock conditions can often be diagnosed more easily.
- Free space on a number of Files-11 devices may be monitored in real time, which may help in situations where on-line storage is at a premium or where some program is running wild.

SHOW MEMORY (Cont.)

Other pages in the RMD display are invoked by the SHOW TASKS/DYNAMIC command (Section 7.12.3.4). Once you have started the RMD display by either SHOW TASKS or SHOW MEMORY, you can move from page to page without issuing any more DCL commands. In fact, RMD is meant to enable you to watch the activity in memory, on the active task list, and within a single task header and to note the relation between these pages. See the RSX-11M/M-PLUS System Management Guide for a full explanation of all the functions of RMD and for descriptions of the commands used within RMD to change from page to page.

CHAPTER 9

QUICK REFERENCE

This chapter lists all DCL commands in alphabetical order. Each command entry includes cross-references to the full command description in this manual or to other RSX-11M manuals. Each command entry also includes a general description of the functions of the command, as well as the full command syntax, sometimes with brief comments behind an exclamation point (!).

These command descriptions are intended for quick reference only. If you are not familiar with the functions of a particular command, you should read the main text referenced in this chapter. The comments are in shorthand form and are meant to serve as reminders only.

Comments include examples of input, value ranges, and cautions. Privileged commands and qualifiers are flagged. A number of commands have two qualifiers listed as synonyms. Other qualifiers perform no operation by themselves and are listed in the comments as "no-ops." These synonyms and no-ops are usually included for compatibility with DCL on VAX/VMS systems or for clarity and completeness. See the main text for more information.

This chapter also lists and summarizes the DCL commands used in setting up the Queue Manager. The RSX-11M/M-PLUS System Management Guide includes full details on using these commands.

For your convenience, some frequently used commands have brief forms. These short forms are not necessarily compatible with other implementations of DCL and are provided for the convenience of RSX-11M users. They are as follows:

A	ABORT	H	HELP
B	BROADCAST	?	HELP
C	COPY	LO	LOGOUT
D	DIRECTORY	L	LINK
DEAL	DEALLOCATE	M	MACRO
DEAS	DEASSIGN	P	PRINT
E	EDIT	R	RUN
F	FORTRAN	S	SHOW

QUICK REFERENCE

ABORT

Section 7.3

ABORT forces an orderly end to a running task or to the action of a specific command. Nonprivileged users can abort any task running on TI:.

Formats

ABORT[/COMMAND][qualifier[s]] commandname

ABORT/TASK[/qualifier] taskname

Command Qualifiers

/COMMAND !Default
/TASK
/[NO]POSTMORTEM

ALLOCATE

Section 5.4

ALLOCATE declares a specified device to be a private device. If you omit the unit number and colon, the first available device of that class is allocated.

Format

ALLOCATE[/qualifier[s]] dd[nn:] [logicalname] !Logical names same as
!other device names:
!Two characters, octal
!unit number, and :.

Command Qualifier

/TERMINAL:ttnn: !Privileged
/TYPE:devicetype !RM05, RK07, RL02, and so on.

ANALYZE/CRASH_DUMP

See the RSX-11M/M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual.

ANALYZE/CRASH_DUMP command helps you determine the cause of system crashes by analyzing and formatting a memory dump created by the Executive Crash Dump Module.

Format

ANALYZE/CRASH_DUMP[/qualifier[s]] filespec[qualifier[s]]

QUICK REFERENCE

Command Qualifiers

```
/LIST[:listfilespec[/qualifiers]]
    /ERROR_LIMIT
    /PAGE_COUNT:n
    /PAGE_LENGTH:n
    /[NO]PRINTER
    /EXIT:n
    /LIMIT:n
    /LINES:n
    /[-]SP
/BINARY:binaryfilespec
/MEMORY_SIZE:n
/SYMBOLS:symbolfilespec
```

File Qualifiers

```
/ACTIVE:(arg[,...])
    DEVICES
    TASKS
/ALL
/DEVICES
/TASKS
/BLOCK:n
/CLOCK_QUEUE
/CONTROLLERS
/DENSITY:n
/DATA_STRUCTURES:(arg[,...])
    COMMAND_PARSER
    DEVICE
    PARTITION
    STATUS
    TASK
    UNIT
/DUMP[(START:n,END:n[,ADDRESS:n])]
/HEADERS
/KERNAL:(arg[,...])
    DATA:(START:n,END:n)
    INSTRUCTION:(START:n,END:n)
    REGISTERS
/PARTITION
/POOL:(START:n,END:n)
/SECONDARY_POOL:(START:n,END:n)
/[NO]SYSTEM
/TASKS:(arg[,...])
    DIRECTORY
    ADDRESS:(NAME:name[,START:n,END:n])
    INSTRUCTION:(NAME:name[,START:n,END:n])
```

ANALYZE/MEDIA

Section 5.2

The ANALYZE/MEDIA command determines if bad blocks exist on a disk volume and records their locations for use by the backup and restore utilities and the INITIALIZE command.

Format

```
ANALYZE/MEDIA[/qualifier[s]] ddnn:
```

QUICK REFERENCE

Command Qualifiers

/ALLOCATE !Prompts for bad block numbers
!to put in BADBLOCK.SYS and to
!enter in the bad block
!descriptor file.

/BAD_BLOCKS
/BAD_BLOCKS/EXERCISE:(n,m)
/BAD_BLOCKS/NOEXERCISE
/[NO]EXERCISE[: (n,m)]
/OVERRIDE
/RETRY
/SHOW

APPEND

Section 4.4.3

APPEND appends to an existing sequential file records from one or more sequential files.

Format

APPEND[/qualifier[s]] infile[,s] outfile

Command Qualifiers

/DATE:dd-mmm-yy !Given day only
/SINCE:dd-mmm-yy !From given day through current day
/THROUGH:dd-mmm-yy !From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy !From given day through given day
/TODAY !Today only
/EXCLUDE:filespec !Filespec can include wildcards.
/NOWARNINGS !Suppress error messages
/REWIND !Tape only. Rewinds tape before
!beginning
/SHARED !Permits others to access file
!while you append it

ASSIGN

Section 5.3

ASSIGN associates a logical name with a physical device, pseudo device, or other logical device. All references to the logical name are resolved by the system to the associated physical device, pseudo device, or logical device. See also other forms of the ASSIGN command described in the following entries.

Format

ASSIGN[/qualifier[s]] ddnn: logicalddnn: !Logical names same as
!other device names:
!Two characters, octal
!unit number, and :.

QUICK REFERENCE

Command Qualifiers

/LOCAL	!Default
/LOGIN	!Privileged
/GLOBAL	!Privileged
/SYSTEM	!Privileged; synonym for /GLOBAL.

ASSIGN/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

ASSIGN/QUEUE establishes a path between a queue and a processor in the Queue Manager subsystem. Privileged.

Format

ASSIGN/QUEUE queueName processorName

ASSIGN/REDIRECT

Section 5.3.1

ASSIGN/REDIRECT redirects output from one physical device to another. You can also redirect output from a physical device to a pseudo device or vice versa. Privileged.

Format

ASSIGN/REDIRECT oldDnn: newDnn:

ASSIGN/TASK

Section 5.3.2

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

Format

ASSIGN/TASK:taskName dDnn: lun

BACKUP

Section 5.10

See also the RSX-11M/M-PLUS Utilities Manual.

BACKUP backs up and restores Files-11 volumes. It transfers files from a volume to a backup volume and retrieves files from the backup volume. BACKUP works through the Backup and Restore Utility (BRU). You should read carefully the main text entries on BACKUP and BRU before using this command.

QUICK REFERENCE

Format

BACKUP[/qualifier[s]] sourceddnn:[filespec[s]] destinationddnn:

Command Qualifiers

Group 1: Selective Backup and Restore

/CREATED:arg
 AFTER:(dd-mmm-yy hh:mm) !Use with /CREATED or
 BEFORE:(dd-mmm-yy hh:mm) !with /MODIFIED.
/EXCLUDE
/IMAGE:arg !For multivolume disk operations.
 SAVE
 RESTORE
/MODIFIED:arg
 AFTER:(dd-mmm-yy hh:mm)
 BEFORE:(dd-mmm-yy hh:mm)
/NEW VERSION
/[NO]REPLACE

Group 2: Initialization

/ACCESSED:n !n is default number of FCBS per
 !volume.
/BADBLOCKS:arg !Default
 AUTOMATIC
 MANUAL
 OVERRIDE
/EXTENSION:n !Default n=5.
/FILE PROTECTION:code !Default is protection of involume.
/HEADERS:n
/INDEX:arg !Specifies location of INDEXF.SYS
 BEGINNING !on volume; default is location of
 MIDDLE !file on involume.
 END
 n !At logical block n.
/INITIALIZE
/MAXIMUM FILES:n
/SAVE_SET:name !Default is name of volume being
 !backed up.
/WINDOWS:n !Default is number of mapping
 !pointers on involume.

Group 3: Tape and Disk Control

/APPEND !May need /REWIND; see main text.
/DENSITY:arg !Default density is 800 bpi.
 800
 1600
/ERROR_LIMIT:n !Default n=25.
/LABEL:TAPE:fileset-ID
/LENGTH:n !Usable length of output tape in
 !decimal feet; default n = physical
 !length of the output tape.
/REWIND !Rewind first tape of tape set
 !before executing the command line;
 !may use with /APPEND.

Group 4: Verification

/COMPARE
/VERIFY

QUICK REFERENCE

Group 5: Display

/LIST
/[NO]LOG !/LOG goes to TI;; /NOLOG is default.

Group 6: Disk Processing

/APPEND !May need /IMAGE; see main text.
/DIRECTORY !Can only be used with /NOINITIALIZE.
/NOINITIALIZE

/LABEL:arg !See also /LABEL:TAPE:volumelabel,
 !Gp 3.
 INPUT:volumelabel
 [OUTPUT:]volumelabel !/LABEL:OUTPUT is default; if the
 !only volumelabel in command line
 !is outvolume, /LABEL:volumelabel
 !will do.

/MOUNTED
/[NO]PRESERVE !/PRESERVE is default.

BROADCAST

Section 3.6

BROADCAST displays a specified message at one or more terminals.

Formats

BROADCAST ttnn: message
BROADCAST @indirectspec
BROADCAST[/qualifier] message

Command Qualifiers

/ALL !Privileged
/LOGGED_IN !Privileged

CANCEL

Section 7.5

CANCEL eliminates entries from the clock queue. CANCEL does not affect a currently executing task, just the pending entries in the clock queue.

Format

CANCEL taskname

CONTINUE

Section 7.4

CONTINUE resumes execution of a previously suspended task.

Format

CONTINUE [taskname] !Taskname defaults to TTnn

QUICK REFERENCE

CONVERT

Section 4.3.6

CONVERT invokes the RMSCNV utility, which moves records from one file to another. RMSCNV reads records from an input file and writes them to an output file. The action of RMSCNV depends on the organization (sequential, relative, or indexed) of the two files and on the qualifiers you include with the CONVERT command. See the main text and the RMS-11 documentation supplied with your system for more information.

Format

CONVERT[/qualifier[s]] infile outfile

Command Qualifiers

/ALLOCATION:n	!Contiguous blocks to allocate; n !with no decimal point is octal
/[NO]APPEND	!/APPEND conflicts with /REPLACE.
/BLOCK_SIZE:n	!Magtape block size:18<=n<=8192; default=512.
/[NO]FIXED_CONTROL	!/NO is default.
/[NO]IDENTIFICATION	!Prints RMS version number; /NO is default; !no filespec required.
/INDEXED	!Outfile is indexed; see main text.
/KEY[:n]	!0<=n<=9; default=1.
/[NO]LOG_FILE[:filespec]	!/NO is default; no filespec logs on TI:
/[NO]MASS_INSERT	
/MERGE	!Both files must have same organization.
/[NO]PAD[:[#]arg]	!Pad infile records to outfile length; default !pad character is blank (040).
/RELATIVE	!Outfile is relative.
/[NO]REPLACE	!/REPLACE conflicts with /APPEND.
/SEQUENTIAL	!Outfile is sequential.
/[NO]TRUNCATE	!/NO is default.

COPY

Section 4.3.4

COPY copies files. Unless specified otherwise, COPY preserves the file organization of the input file, that is, indexed files are copied as indexed files, and so forth. See also the CONVERT command.

Format

COPY infile[s] outfile !See main text for multiple file copies

Command Qualifiers

/ALLOCATION:n	!Number of contiguous block space
/BLOCK_SIZE:n	!Defines block size for outfile on !magtape; n is octal unless terminated !with decimal point; no effect on !infile.
/[NO]CONTIGUOUS	!Specifies contiguous outfile.
/DATE:dd-mmm-yy	!Given day only
/EXCLUDE:filespec	!File specification can !include wildcards.

QUICK REFERENCE

/NOWARNINGS	!Suppresses error messages
/OWN	!Makes outfile UIC owner of copy.
/OVERLAY	!Infile written over outfile.
/PRESERVE_DATE	!Output file takes creation date of input file.
/REPLACE	
/REWIND	!Magnetic tape only. Rewinds tape before starting operation.
/SHARED	!Permits other to access file while you copy it.
/SINCE:dd-mmm-yy	!From given day through current day
/THROUGH:dd-mmm-yy	!From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy	!From given day through given day
/TODAY	!Today only

CREATE

Section 4.2.1

CREATE creates a sequential file in a directory on a file-structured device. After you issue the command, you can immediately enter text. To end data entry and close the file, type CTRL/Z. See also CREATE/DIRECTORY.

Format

CREATE filespec

CREATE/DIRECTORY

Section 4.2.2

CREATE/DIRECTORY creates a User File Directory (UFD) on a Files-11 volume and enters its name in the volume's Master File Directory (MFD). Nonprivileged users can only create directories on mounted volumes on their own private (allocated) devices.

Format

CREATE/DIRECTORY[/qualifier] [ddnn:] [g,m]

Command Qualifiers

/ALLOCATION:n	!Entries for n files.
/LABEL:volumelabel	!Compare with volume label.
/NOWARNINGS	!Suppresses error messages
/OWNER_UIC	!Owner of directory; use if different from creator.
/PROTECTION:code	

DEALLOCATE

Section 5.6

DEALLOCATE counteracts ALLOCATE. It frees a private device for access by others.

QUICK REFERENCE

Format

DEALLOCATE[/qualifier] ddnn: !Volume must be dismounted.

Command Qualifier

/ALL !Frees all devices allocated by TI;;
!do not specify ddnn:.
/DEVICE !No-op

DEASSIGN

Section 5.4

DEASSIGN deletes logical-device assignments. DEASSIGN disassociates logical names from physical device names, pseudo device names, or logical device names created by ASSIGN. See also other forms of the DEASSIGN command described in the following entries.

Format

DEASSIGN[/qualifier[s]] logicalddnn:

Command Qualifiers

/ALL !Combine with any other qualifier;
!No ddnn:
/LOCAL !Default
/LOGIN !Privileged
/GLOBAL !Privileged
/SYSTEM !Synonym for /GLOBAL.

DEASSIGN/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

DEASSIGN/QUEUE counteracts ASSIGN/QUEUE. It is used to eliminate the path from a queue to a processor in the Queue Manager subsystem. Privileged.

Format

DEASSIGN/QUEUE queuename processorname

DELETE

Section 4.3.2

DELETE deletes specified versions of files and releases the storage space the files occupy. See also other forms of the DELETE command described in the following entries.

Format

DELETE[/qualifier[s]]

QUICK REFERENCE

Command Qualifiers

/[NO]CONFIRM	!Synonym to /QUERY
/[NO]LOG	!Lists deleted files on TI:
/[NO]QUERY	!Queries before deleting;
	!/NO is default.
/NOWARNINGS	!Suppresses error messages
/DATE:dd-mmm-yy	!Given day only
/SINCE:dd-mmm-yy	!From given day through current day
/THROUGH:dd-mmm-yy	!From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy	!From given day through given day
/TODAY	!Today only
/EXCLUDE:filespec	!Filespec can include wildcards.

DELETE/ENTRY

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

DELETE/ENTRY deletes QMG jobs by entry number.

Format

DELETE/ENTRY:n[/qualifier] !n is QMG entry number.

Command Qualifier

/FILE_POSITION:n

DELETE/JOB

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

DELETE/JOB deletes QMG jobs by queue name and job name.

Format

DELETE/JOB[/qualifier] queue name [[g,m]]job name

Command Qualifier

/FILE_POSITION:n

DELETE/processor type

See the RSX-11M/M-PLUS System Management Guide.

DELETE/PROCESSOR deletes print processors, output despoolers, or batch processors from the Queue Manager subsystem by processor name or device name. This command also sets the device unspooled.

QUICK REFERENCE

Format

DELETE/processorname

APPLICATIONS PROCESSOR	
BATCH PROCESSOR	
CARD READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

DELETE/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

Privileged. DELETE/QUEUE deletes queues in the Queue Manager subsystem by name. See DELETE/JOB and DELETE/ENTRY to delete jobs from queues.

Format

DELETE/QUEUE queue/ERASE !/ERASE is required.

DIFFERENCES

Section 4.4.4

DIFFERENCES compares two ASCII (text) files line by line to determine if parallel records (lines) are identical and produces a listing of the differences between the files, if any.

Format

DIFFERENCES infile1 infile2

Command Qualifiers

/CHANGE_BAR[:n]	!n is octal ASCII code of alternate !bar. Default n=041 (!).
/IGNORE:(arg[,s])	
BLANK LINES	
COMMENTS	!Comments begin with ";".
FORM FEEDS	
SPACING	!Group of tabs and blanks = 1 blank.
TRAILING BLANKS	
/LINES:n	!n lines must be same for match.
/[NO]NUMBERS	!Line numbers in output file.
/OUTPUT:filespec	!Names output file; TI: is default.
/SLP[:audittrail]	!Output is SLP indirect command file.

QUICK REFERENCE

DIRECTORY

Section 4.3.1

DIRECTORY displays information on files in directories (UFDs).

Format

DIRECTORY [/format-qual] [/destination-qual] [other-qual[s]] [filespec[s]]

Command Qualifiers

Format Qualifiers

/ATTRIBUTES	!RMS-11 attributes
/BRIEF	
/FREE [ddnn:]	!Free blocks on volume; default is !SY:.
/FULL	
/SUMMARY	!Blocks used and allocated only.

Destination Qualifier

/OUTPUT[:filespec]	!Names output file. TI: is default.
--------------------	-------------------------------------

Other Qualifiers

	!Not used with /FREE or /ATT
/DATE:dd-mmm-yy	!Given day only
/SINCE:dd-mmm-yy	!From given day through current day
/THROUGH:dd-mmm-yy	!From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy	!From given day through given day
/TODAY	!Today only
/EXCLUDE:filespec	!File specification can include wildcards.
/NOWARNINGS	!Suppresses error messages
/REWIND	!Magnetic tape only; Rewinds tape before starting operation.

DISMOUNT

Section 5.8

DISMOUNT marks the volume mounted on the specified device as logically off line and disconnected from the file system.

Format

DISMOUNT ddnn: [label]	!If label specified, volume label is !checked.
------------------------	--

Command Qualifiers

/ALL	!Dismount all devices mounted by user.
/PUBLIC	!Privileged; dismounts all users from !volume.
/SAVE	!Privileged; disk keeps spinning; !privileged tasks can access.
/SYSTEM	!Synonym for /PUBLIC.
/TERMINAL	!Privileged; dismounts volumes !mounted from another terminal
/[NO]UNLOAD	!Magnetic tape only.
	!See main text.

QUICK REFERENCE

EDIT[/EDT]

Section 4.2.3.1

See also the EDT Editor Manual.

EDIT or EDIT/EDT invokes EDT, the DEC Standard Editor. EDT is the default editor.

Format

EDIT[/EDT][/qualifier[s]] filespec

Command Qualifiers

/[NO]COMMAND[:filespec]	!Default is /COMMAND:EDTINI.EDT.
/[NO]CREATE	
/[NO]JOURNAL[:filespec]	
/[NO]OUTPUT[:filespec]	
/[NO]READ ONLY	!Default is /NOREAD ONLY.
/[NO]RECOVER	!Default is /NORECOVER.

EDIT/SLP

Section 4.2.3.2

See also the RSX-11M/M-PLUS Utilities Manual.

EDIT/SLP invokes the Source Language Input Program (SLP), a program-maintenance editor.

Format

EDIT/SLP[/qualifier[s]] filespec

Command Qualifiers

/[NO]AUDIT[:(<u>arg[s]</u>)]	!Default is /AUDIT:(POS:80:SIZE:8).
POSITION:n	!n must be <=132.
SIZE:n	!n must be <14.
/CHECKSUM[:n]	
/[NO]LIST[:filespec]	
/[NO]OUTPUT[:filespec]	
/[NO]REPORT	!Report truncation of lines by audit !trail.
/[NO]TAB	!Right-justify with tabs or spaces; !/NOTAB is default.
/[NO]TRUNCATE[:n]	!/TRUNCATE deletes audit trails and !trailing characters; optionally !truncates text at specified !horizontal position.

EDIT/EDI and EDIT/editor

Section 4.2.3.3

EDIT/EDI invokes EDI, the Line Text Editor. EDIT/editor invokes a user-specified editor (see list below).

QUICK REFERENCE

Format

EDIT[/qualifier] [edit-input] !Input accepted by editor

Command Qualifier

/EDI	!Line Text Editor.
/KED	!Unbundled KED editor.
/K52	!VT52 version of unbundled KED editor.
/OUTPUT:filespec	!Use with /KED and /K52 (unbundled editors).
/CREATE	!Use with KED and K52.
/SOS	!Unsupported Son of Stopgap.
/TECO	!Unsupported Text Editor and Corrector.
/USING:yyy	!Unsupported user editor.

FIX

Section 7.10

FIX causes an installed task or region to be loaded and locked into memory. Privileged.

Format

FIX taskname !Name task

HELP

Section 3.5

Help displays information about your system. Help files for MCR, DCL, and most utilities are supplied with the system. Your system may also have help for an alternate CLI or local, group, or other special help. See main text.

Format

HELP[/qualifier[s]] [parameter1 ...parameter9]

Command Qualifiers

/OUTPUT:filespec	!Help text to file; default /OUTPUT:TI:.
/LOCAL	!Help file is in default UFD; HELP % is same.
/GROUP	!Help file is in [g,l]; g is your group.
/CLI:cliname	!For alternate CLIs.
/MCR	!MCR help; default for MCR terminals.
/DCL	!DCL help; default for DCL terminals.
/FILE:filespec	!Names file containing help text.
/filename	!LB: [1,2] filename.HLP

QUICK REFERENCE

HOLD/ENTRY

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

HOLD/ENTRY holds a QMG job in its queue by entry number.

Format

HOLD/ENTRY:n !n is QMG entry number.

HOLD/JOB

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

HOLD/JOB holds a QMG job in its queue by queuename and jobname.

Format

HOLD/JOB queuename [[g,m]]jobname

INITIALIZE

Section 5.9

INITIALIZE produces a volume in Files-11 format. See also INITIALIZE/UPDATE. You can mount the volume /FOREIGN, but it is not required. See main text. See also other forms of the INITIALIZE command described in the following entries.

Format

INITIALIZE[/qualifier[s]] ddn: volumelabel !Nonprivileged must
 !allocate the device.

Command Qualifiers

/ACCESSED:n !Number of directories accessed
 !simultaneously.

/BAD_BLOCKS:arg !Default
 AUTOMATIC
 (AUTOMATIC,MANUAL)
 MANUAL
 NOAUTOMATIC
 OVERRIDE
 (OVERRIDE,MANUAL)

/DENSITY:arg
 800
 1600
 HIGH
 LOW

/EXTENSION:n !Extend full files by n blocks;
 !default n=5.

QUICK REFERENCE

/FILE_PROTECTION: (code) !Default protection for files
!on volume.

/HEADERS:n

/INDEX:arg !Locates index file on volume.
BEGINNING !Default for tapes and DECTapes.
MIDDLE !Default for disks.
END
n !At logical block n.

/LABEL:VOLUME_ACCESSIBILITY:"c"
!Magnetic tape only; limits
!access; c can be A--Z, 0--9,
!and ! " % ' () + , - / :
!= & > < _ ? or ;

/MAXIMUM FILES:n

/OWNER: [g,m]

/PROTECTION: (code) !Default protection for volume.

/[NO]SHOW !Display volume information on TI;;
!/NO is default.

/WINDOWS:n !Mapping pointers to file windows;
!default n=7.

INITIALIZE/PROCESSOR (Output)

See the RSX-11M/M-PLUS System Management Guide.

INITIALIZE/PROCESSOR creates, names, and starts an output despooler or batch processor. Privileged. See next entry to initialize an input spooler.

Format

INITIALIZE/processor_{type} processor_{name}[/qualifier[s]]

Processor_{types}:

APPLICATIONS_PROCESSOR
BATCH_PROCESSOR
DEVICE
PRINTER
PROCESSOR

Command Qualifiers

/FLAG_PAGE:n !n can be 0-2; default n=1.

/FORMS:n !n can be 0-255; default n=0.

/[NO]SHAREABLE !Default is NOSHAREABLE.

/[NO]LOWERCASE !Default is LOWERCASE.

/[NO]UPPERCASE !Default is NOUPPERCASE.

INITIALIZE/PROCESSOR (Input)

See the RSX-11M/M-PLUS System Management Guide.

INITIALIZE/processor_{type} creates, names, and starts an input spooler or card-reader processor. Privileged. See previous entry to initialize an output despooler.

QUICK REFERENCE

Format

```
INITIALIZE/processortype processorname[/qualifier[s]]
      CARD_READER      !INPUT is synonym.
      INPUT            !CARD_READER is synonym.
```

Command Qualifiers

```
/CONSOLE:ddnn:
/PRINTER_QUEUE:queuename
```

INITIALIZE/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

INITIALIZE/QUEUE creates, names, and starts a queue in the Queue Manager subsystem.

Format

```
INITIALIZE/QUEUE queuename[/qualifier]
```

Command Qualifiers

```
/PRINTER                !Default
/NOWARNINGS
```

INITIALIZE/UPDATE

Section 5.9.1

INITIALIZE/UPDATE invokes the HOME utility to alter values in the volume home block without affecting the other data on the volume. INITIALIZE/UPDATE is only for disks and DECTapes in Files-11 format.

You can mount the volume /FOREIGN on RSX-11M, but it is not required. See main text.

Format

```
INITIALIZE/UPDATE[/qualifier[s]] ddnn: volumelabel
```

Command Qualifiers

```
/ACCESSED:n
/DENSITY:arg
      HIGH
      LOW
/EXTENSION:n                !Extend full files
                              !by n blocks.
/FILE_PROTECTION:code       !Default protection for files
                              !on volume.
/LABEL:newvolumelabel      !Changes volume label.
/MAXIMUM_FILES:n           !Maximum number of files
                              !on volume.
/OWNER:[g,m]
/PROTECTION:code           !Protection for volume.
/[NO]SHOW
/WINDOWS:n                 !Mapping pointers to file
                              !windows; default n=7.
```

QUICK REFERENCE

INSTALL

Section 7.8

INSTALL includes a task in the System Task Directory, thus making it known to the system. Privileged.

Format

INSTALL[/qualifier[s]] [\$]filespec

Command Qualifiers

/[NO]CHECKPOINT	
/COMMAND:"taskcommand"	
/EXTENSION:n	!n (decimal) additional words !of address space.
/[NO]INTERPRETER	!Installing a CLI? Default is no.
/MULTIUSER_PARTITION:parname	!Install read-only portion.
/PARTITION:parname	
/[NO]POSTMORTEM	
/PRIORITY:n	!0-250.
/READONLY_COMMON	!Install common as read-only.
/[NO]SLAVE	!/NOSLAVE is default.
/TASK_NAME:taskname	!1-6 characters.
/UIC:[g,m]	

LIBRARY

Section 6.3

See also the RSX-11M/M-PLUS Utilities Manual.

LIBRARY creates and maintains user-written library files. The command has eight functions, each listed here as a separate command. See main text for more details on all functions and qualifiers.

Format

LIBRARY[/operation][qualifier[s]]

LIBRARY @filespec !Indirect command file specification.

LIBRARY/COMPRESS

Section 6.3.1

LIBRARY/COMPRESS physically deletes modules that have been logically deleted through LIBRARY/DELETE. You can rename the resulting compressed library. You can also use this command to copy a library and rename it.

QUICK REFERENCE

Format

```
LIBRARY/COMPRESS[: (arg[,s])] libspec [newlibspec]
                GLOBAL:n      !Entry-point table entries.
                MODULES:n     !Module-name table entries.
                BLOCKS:n      !Size in 256-word blocks.
```

LIBRARY/CREATE

Section 6.3.2

LIBRARY/CREATE creates a library and optionally inserts one or more modules into it.

Format

```
LIBRARY/CREATE[: (arg[,s])][ /qualifier[s]] libspec [infilespec[s]]
                GLOBAL:n      !Entry-point table entries.
                MODULES:n     !Module-name table entries.
                BLOCKS:n      !Size in 256-word blocks.
```

Command Qualifiers

```
/[NO]GLOBALS      !Include globals in entry-point table.
/MACRO            !Identifies macro library.
/OBJECT          !Default; identifies object library.
/SELECTIVE_SEARCH !Object modules only.
/SQUEEZE
/UNIVERSAL       !Identifies universal library.
```

LIBRARY/DELETE

Section 6.3.3

LIBRARY/DELETE deletes object modules from a library. See LIBRARY/REMOVE for removing global symbols (entry points) from a library.

Format

```
LIBRARY/DELETE libspec module[,module[,s]] !Name as many as 15 modules
```

LIBRARY/EXTRACT

Section 6.3.4

LIBRARY/EXTRACT reads one or more modules from a library and writes them to a specified output file. You can extract as many as eight modules with a single command. If you extract more than one module, the modules are concatenated in the output file.

Format

```
LIBRARY/EXTRACT[/qualifier] libspec module[,s]
```

QUICK REFERENCE

Command Qualifier

/OUTPUT[:filespec] !Default output file is TI; name
!a file when extracting object
!modules.

LIBRARY/INSERT

Section 6.3.5

LIBRARY/INSERT inserts modules from one or more files into a library.

Format

LIBRARY/INSERT libspec filespec[s] !Name any number of files

Command Qualifiers

/[NO]GLOBALS !Include globals in entry-point table.
/SELECTIVE_SEARCH
/SQUEEZE

LIBRARY/LIST

Section 6.3.6

LIBRARY/LIST lists on your terminal or in an output file the names of all modules in a library.

Format

LIBRARY/LIST[:filespec] libspec

Command Qualifiers

/BRIEF
/FULL
/[NO]NAMES !Names plus global entry points.

LIBRARY/REMOVE

Section 6.3.7

LIBRARY/REMOVE removes global symbols (entry points) from a library. See LIBRARY/DELETE for deleting object modules from a library.

Format

LIBRARY/REMOVE libspec global[,global[,s] !Name as many as !15 global symbols

QUICK REFERENCE

LIBRARY/REPLACE

Section 6.3.8

LIBRARY/REPLACE replaces a module in a library with a new module of the same name and deletes the old module.

Format

LIBRARY/REPLACE libspec filespec[s]

Command Qualifiers

/[NO]GLOBALS !Include globals in entry-point table.
/SELECTIVE_SEARCH
/SQUEEZE

LINK

Section 6.2.2

LINK invokes the Task Builder, which links object modules and routines from user and system libraries to form an executable task.

LINK[/qualifier[s]] filespec[/qualifier[s]][,filespec[s]]

Command Qualifiers

/ANCILLARY_PROCESSOR[:n] !Task is ACP; n is 0,4, or 5.
/[NO]CHECKPOINT:arg !Checkpoints to [0,0]CORIMG.SYS.
 SYSTEM !Checkpoints to task image file.
 TASK
/CODE:(arg[,s]) !Extended arithmetic element.
 EAE !Floating-point processor.
 [NO]FPP !Position-independent code.
 PIC !Synonym for PIC.
 POSITION_INDEPENDENT
/COMPATIBLE
/[NO]CROSS_REFERENCE
/[NO]DEBUG[:filespec] !Default is ODT.
/[NO]EXECUTABLE:filespec !Names task file. Synonym for /TASK.
/ERROR_LIMIT:n !Stop task build after n errors.
/FAST !Fast TKB.
/FULL_SEARCH
/[NO]HEADER
/[NO]IO_PAGE
/LONG !Long map
/MAP[:filespec] !Default is /MAP:TI:
/[NO]MEMORY_MANAGEMENT[:n] !n for unmapped systems; see main
 !text; default is /MEM; n is 28 (default) or 30.
/OPTIONS[:filespec] !File contains options. Otherwise you will be prompted
/OVERLAY_DESCRIPTION
/POSTMORTEM
/[NO]PRINT !Print map file?
/[NO]PRIVILEGED[:n] !Default is /NOPRIVILEGED.
/[NO]RECEIVE
/[NO]RESIDENT_OVERLAYS

QUICK REFERENCE

`/SAVE` !Saves indirect file.
`/SELECTIVE_SEARCH`
`/[NO]SEGREGATE` !/NOSEG is default.
`/SEQUENTIAL`
`/SHAREABLE[:arg]` !Multiuser
 COMMON
 LIBRARY
`/SLAVE`
`/SLOW` !Less space consuming TKB
`/SYMBOL_TABLE[:filespec]` !Output .STB file
`/[NO]SYSTEM LIBRARY DISPLAY` !Default is /NOSYS.
`/[NO]TASK[:filespec]` !Names task image file; /EXECUTABLE
 !is synonym; names 1-6 characters;
 !/NOTASK means no task built.
`/TKB` !Default
`/TRACE`
`/[NO]WARNINGS` !/NO suppresses diagnostic messages; /WARNINGS is
 !default.
`/WIDE` !Wide map

File Qualifiers

`/[NO]CONCATENATE`
`/DEFAULT LIBRARY` !File to replace [1,1][SYSLIB.OLB
`/[NO]GLOBALS` !Default is /GLOBALS; includes
 !global symbols in map.
`/LIBRARY` !File is object module library.
`/INCLUDE:(module1[:...:modulen])` !File is object module library;
 !include named modules in task image.
`/OVERLAY_DESCRIPTION` !File is .ODL; also a command qualifier.
`/SELECTIVE_SEARCH` !Also a command qualifier.

LOGIN

Section 3.1.1

LOGIN (or HELLO) grants access to a multiuser protection system and establishes your privileges as a system user.

LOGIN userid password !HELLO is synonym.

LOGOUT

Section 3.1.2

LOGOUT counteracts LOGIN. LOGOUT also aborts any nonprivileged tasks running from the terminal and also dismounts any volumes and deallocates any private devices allocated from the terminal.

Format

LOGOUT[/qualifier] !LO is sufficient

Command Qualifier

`/[NO]HOLD` !Holds remote line after logout; /NO
 !is default.

QUICK REFERENCE

MCR

Section 1.2.4

MCR enters an MCR command from a DCL terminal without leaving DCL.

Format

MCR mcrcommand

MOUNT

Section 5.7

MOUNT declares a volume to be logically known to the system, on line, and available for use. Some qualifiers can be used with any MOUNT command; some are limited to mounting disks (and other random-addressable devices) and others are limited to mounting magnetic tapes.

Format for Disks and Other Random-Addressable Devices

MOUNT[/qualifier[s]] ddnn: volumelabel

Format for Magnetic Tapes

MOUNT[/qualifier[s]] ddnn:[,ddnn:...] fileset-ID

Command Qualifiers for Both Disks and Tapes

/FILE PROTECTION:(code)	!Protection for files created during mount
/FOREIGN	
/OVERRIDE:IDENTIFICATION	!Privileged; no label needed.
/PARAMETERS:"user parameters"	!Quotes are required syntax.
/PROCESSOR:arg	!Privileged; name ACP for volume.
acpname	
UNIQUE	
/PROTECTION:(code)	!Volume protection during mount
/PUBLIC	!Privileged; deallocates; sets public.
/[NO]SHAREABLE	
/[NO]SHOW	!Display volume information on TI:.
/SYSTEM	!Synonym for /PUBLIC.
/[NO]WAIT	!Default is /NOWAIT.
/[NO]WRITE	

Command Qualifiers for Files-11 Devices

/ACCESSED:n	!n is number of File Control Blocks.
/EXTENSION:n	!Extend full files by n blocks.
/OWNER:[g,m]	!Coordinate with file and volume protection
/UNLOCK	!Main use is with VFY.
/WINDOW:arg	
n	
(USER:n,INDEX:n)	
FULL	

Command Qualifiers for ANSI and Unlabeled Tapes

See main text for these qualifiers.

/BLOCK SIZE:n	
/CARRIAGE_CONTROL:arg	
FORTRAN	
LIST	
NONE	

QUICK REFERENCE

```
/DENSITY:arg
      800
      1600
/[NO]HDR3
/[NO]LABEL
/OVERRIDE:(arg[,s])
      ACCESSIBILITY
      EXPIRATION_DATE
      SET_IDENTIFICATION
/RECORD_SIZE:n
/TRANSLATE:arg
      EBCDIC
      NONE
      UT1
      UT2
      UT3
/VOLUME_IDENTIFICATION:(volume-ID[,volume-ID[,s]])
```

PRINT

Section 4.4.2

PRINT queues files for printing on a line printer. PRINT can also queue jobs for other output devices.

Format

```
PRINT[/qualifier[s]] filespec[/qualifier[s]][,filespec[,s]]
```

Command Qualifiers

```
/AFTER:TOMORROW
/AFTER:(dd-mmm-yy hh:mm)
/COPIES:n !Override on filespec.
/[NO]DELETE !Override on filespec.
/DEVICE:ddnn:
/[NO]FLAG_PAGE !Flag page on each file; default is
!/NOFLAG_PAGE.
/FORMS:n !n can be 0-256; default n=0.
/[NO]HOLD !Default is /NOHOLD; same effect as
!HOLD command.
/JOB_COUNT
/[NO]JOB_PAGE !Flagpage on job; default /JOB_PAGE.
/LENGTH:n !Page length
/[NO]LOWERCASE
/NAME:jobname !1-9 characters
/PAGE_COUNT:n !Limit pages in job.
/PRIORITY:n !n is 1-150 nonprivileged; through 250
!privileged; default n=50.
/QUEUE:queuname
/[NO]RESTART
/[NO]UPPERCASE
/NOWARNINGS !Suppresses error messages.
```

File Qualifiers

```
/COPIES:n
/[NO]DELETE
```


QUICK REFERENCE

PURGE

Section 4.3.3

PURGE deletes all but the latest versions of files and releases the storage space the deleted files occupy.

Format

PURGE[/qualifier[s]] filespec[s]

Command Qualifiers

/KEEP:n	
/[NO]LOG	!Lists files on TI: as deleted.
/NOWARNINGS	!Suppresses error messages
/DATE:dd- <u>mmm</u> -yy	!Given day only
/SINCE:dd- <u>mmm</u> -yy	!From given day through current day
/THROUGH:dd- <u>mmm</u> -yy	!From beginning through given day
/SINCE:dd- <u>mmm</u> -yy/THROUGH:dd- <u>mmm</u> -yy	!From given day through given day
/TODAY	!Today only
/EXCLUDE:filespec	!File specification can include wildcards.

RELEASE/ENTRY

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

RELEASE/ENTRY releases by entry number a print or batch job that has been held in its queue.

Format

RELEASE/ENTRY:n !n is QMG entry number.

RELEASE/JOB

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

RELEASE/JOB releases by queue name and job name a print or batch job that has been held in its queue.

Format

RELEASE/JOB queue name [[g,m]]job name

REMOVE

Section 7.9

REMOVE counteracts INSTALL. REMOVE takes a task name out of the System Task Directory. Privileged.

Format

REMOVE taskname

QUICK REFERENCE

RENAME

Section 4.3.5

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

Format

```
RENAME[/qualifier[s]] oldfilespec newfilespec      !Watch wildcards
                                                    !and version
                                                    !numbers.
```

Command Qualifiers

```
/NEW_VERSION          !Supplies new version
                      !number to outfile.
/NOWARNINGS           !Suppresses error messages
/DATE:dd-mmm-yy       !Given day only
/SINCE:dd-mmm-yy      !From given day through
                      !current day
/THROUGH:dd-mmm-yy    !From beginning through
                      !given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy !From given day
                      !through given day
/TODAY                !Today only
/EXCLUDE:filespec     !File specification
                      !can include wildcards.
```

REQUEST

Section 3.7

REQUEST sends a message to the operator's console (CO:).

Format

```
REQUEST message          !Optional "quotes" around message
```

RUN uninstalled task

Section 7.2.1

When used to run an uninstalled task (from a task image file), RUN is a combination command, encompassing INSTALL, RUN, and REMOVE. See main text.

Format

```
RUN[/qualifier[s]] [extract_itex]filespec
```

QUICK REFERENCE

Command Qualifiers

/[NO]CHECKPOINT	
/COMMAND:"taskcommand"	!See main text.
/EXTENSION:n	!n (octal) additional words of address !space.
/[NO]IO PAGE	
/PARTITION:parname	!Privileged
/[NO]POSTMORTEM	
/PRIORITY:n	!Privileged
/[NO]SLAVE	
/STATUS:arg	
TASK	
COMMAND	
/TASK NAME:taskname	!1-6 characters
/UIC:[g,m]	!Privileged.

RUN installed task

Section 7.2.2

RUN also initiates the execution of installed tasks. Privileged users can use RUN to initiate the execution of installed tasks on a schedule by creating entries in the system clock queue.

Format

RUN[/qualifier[s]] taskname

Command Qualifiers	!All privileged except /STATUS
/DELAY:nu	!n is the number of units and u is the !time unit: T,ticks; S,seconds; !M,minutes; H,hours.
/INTERVAL:nu	
/SCHEDULE:hh:mm:ss	
/STATUS:arg	!Can be used only with /UIC:[g,m]
COMMAND	!Return status from RUN command; default.
TASK	!Return status from task being run; !see main text on both these !arguments.
/SYNCHRONIZE:u	!Synchronize on next T, S, M, or H.
/UIC:[g,m]	

SET [DAY]TIME

Section 8.1.1

SET [DAY]TIME sets the system date and time. Privileged.

Format

SET [DAY]TIME:[dd-mmm-yy] [hh:mm] !Set at least one; either order.

QUICK REFERENCE

SET DEBUG

Section 1.2.15

SET DEBUG displays MCR translation of any DCL command.

Format

SET DEBUG[/qualifier[s]]

Command Qualifiers

/[NO]EXECUTE	!Indicates whether or not to execute !the DCL command after translation.
/FULL	!Displays logical symbols.

SET DEFAULT

Section 4.5.1

SET DEFAULT establishes your default device or directory, or both.

Format

SET DEFAULT [ddnn:][[g,m]]	!Unadorned SET DEF returns !nonprivileged user to login device !and directory.
----------------------------	--

SET DEVICE

Section 5.10.2

SET DEVICE establishes certain device attributes. Privileged.

Format

SET DEVICE:ddnn:/qualifier[s]

Command Qualifiers

/[NO]CHECKPOINT_FILE[:n]	!n is number (decimal) of blocks in ![0,0]CORIMG.SYS.
/[NO]LOWERCASE	!Default is /NOLOWERCASE.
/[NO]PUBLIC	!Default is /NOPUBLIC.
/[NO]SYSTEM	!Synonym for /[NO]PUBLIC
/WIDTH:n	!Nonprivileged for TI:.

SET FILE

Section 4.5.4

SET FILE establishes certain file attributes.

Format

SET FILE[/qualifier[s]] filespec[s]

QUICK REFERENCE

Command Qualifiers

/END_OF_FILE: (BLOCK:n, BYTE:n)	!See main text.
/ENTER:synonym_filespec	!See main text.
/NOWARNINGS	!Suppresses error messages.
/REMOVE	!See main text.
/REWIND	!Magtape only; rewinds tape before starting operation.
/TRUNCATE	!Eliminates blocks allocated but unused; saves disk space.

SET GROUPFLAGS

Section 6.4.1

SET GROUPFLAGS creates and deletes group global event flags. See main text.

Format

SET GROUPFLAGS:n[/qualifier] !Nonprivileged command for own group
!n is group number.

Command Qualifiers

/CREATE	!Default
/DELETE	

SET HOST

Section 1.2.12.1

SET HOST connects your terminal to a remote system. You issue this command after you have logged in on your current system. Both your current and remote system must run DECnet software.

SET HOST nodename	!nodename is the name of the remote system.
-------------------	---

SET [NO]PARTITION

Section 7.12.1

SET [NO]PARTITION creates or eliminates a partition. Privileged.

Format

SET [NO]PARTITION:parname/qualifier[s] !Parname 1-6 characters

QUICK REFERENCE

Command Qualifiers

/BASE:n
/COMMON
/DEVICE !Device common.
/SIZE:n
/[NO]SUBPARTITION:subparname
/SYSTEM
/TASK
/TOP:value

SET PRIORITY

Section 7.12.4

SET PRIORITY alters the priority of an active task. Privileged.

Format

SET PRIORITY:n taskname !n = 0 through 250

SET PROTECTION

Section 4.5.3

SET PROTECTION establishes the protection status of files.

Format

SET PROTECTION:code[/qualifier[s]] filespec[s] !The preferred syntax;

SET PROTECTION[/qualifier[s]] filespec[s](code) !default is (SY:RWED,OW:RWED,GR:RWED,W:R).

Command Qualifiers

/DATE:dd-mmm-yy !Given day only
/SINCE:dd-mmm-yy !From given day through current day
/THROUGH:dd-mmm-yy !From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy !From given day through given day
/TODAY !Today only
/EXCLUDE:filespec !Filespec can include wildcards.

SET QUEUE/ENTRY

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

SET QUEUE/ENTRY modifies by entry number some attributes of print or batch jobs once they are in a queue. See SET QUEUE/JOB to modify by job name.

Format

SET QUEUE/ENTRY:n[/qualifier] !Only one attribute at a time

QUICK REFERENCE

Command Qualifiers

/AFTER:(dd-mmm-yy hh:mm)
/COPIES:n
/[NO]DELETE
/FILE POSITION:n
/FORMS:n
/HOLD !Same as HOLD/QUEUE
/LENGTH:n
/[NO]LOWERCASE
/PAGE COUNT:n
/PRIORITY:n !n is 1-150 nonprivileged; through 250
!privileged; default n=50.
/RELEASE !Same as RELEASE/QUEUE
/[NO]RESTART
/[NO]UPPERCASE

SET QUEUE/JOB

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

SET QUEUE/JOB modifies by job name some attributes of print or batch jobs once they are in a queue. See SET QUEUE/ENTRY to modify by entry number.

Format

SET QUEUE/JOB[/qualifier] queuename [[g,m]]jobname
!Only one attribute at a time

Command Qualifiers

/AFTER:(ddd-mmm-yy hh:mm)
/COPIES:n
/[NO]DELETE
/FILE POSITION:n
/FORMS:n
/HOLD !Same as HOLD/QUEUE
/LENGTH:n
/[NO]LOWERCASE
/PAGE COUNT:n
/PRIORITY:n !n is 1-150 nonprivileged; through 250
!privileged; default n=50.
/RELEASE !Same as RELEASE/QUEUE
/[NO]RESTART
/[NO]UPPERCASE

SET SYSTEM

Section 8.1.3

SET SYSTEM establishes certain characteristics of the system. It is a privileged command.

Format

SET SYSTEM/qualifier !Qualifier required

QUICK REFERENCE

Command Qualifiers

/DIRECTORY:[g,m]	!Sets directory where system tasks are kept; does not create a directory.
/EXTENSION_LIMIT:n	!Sets maximum size a task can extend itself with the Extend Task directive.
/[NO]LOGINS	
/NETWORK_UIC:[g,m]	!Sets UIC for network software
/PACKETS:n	!n = 0 through 15
/POOL:top:max:total	!Increases size of pool; see main !text.
/POOL/LIMITS	!Sets pool limit parameters

SET TERMINAL

Section 3.8.1

SET TERMINAL sets various attributes of your terminal. Privileged users can set attributes for any terminal.

Format

SET TERMINAL[:ttnn:]/qualifier[s]

Command Qualifiers

Group 1: Common Use

/[NO]BROADCAST	
/CLI:cliname	
/DCL	
/[NO]HOLD_SCREEN	!Not for VT100s
/INQUIRE	
/[NO]LOWERCASE	!/NOLOWER same as /UPPER and is !default.
/MCR	
/[NO]PRIVILEGED	!Privileged
/SPEED:(transmit,receive)	!Remember to set hardware after this !command.
/[NO]UPPERCASE	!/NOUPPERCASE same as /LOWERCASE
/WIDTH:n	

Group 2: Terminal Setup

/[NO]ADVANCED_VIDEO	
/[NO]ANSI CRT	
/[NO]AUTOBAUD	
/ASR33	
/ASR35	
/[NO]BLOCK_MODE	
/CRFILL:n	!n can be 0-7
/DEC CRT	
/DTC01	
/EDIT MODE	
/[NO]FORM FEED	
/[NO]HARDCOPY	
/HOSTSYNC	
/KSR33	
/LA12	

QUICK REFERENCE

/LA24
/LA30P
/LA30S
/LA34
/LA36
/LA38
/LA50
/LA100
/LA120
/LA180S
/LA210
/LFFILL
/LN03
/LQP02
/LQP03
/MODEL:arg
/PAGE LENGTH:n
/[NO]PRIORITY[:n]
/PRINTER PORT
/PRO SERIES
/[NO]REGIS
/[NO]SCOPE
/[NO]SOFT CHARACTERS
/[NO]TAB
/[NO]TTSYNC
/VT05
/VT50
/VT52
/VT55
/VT61
/VT100
/VT101
/VT102
/VT105
/VT125
/VT131
/VT132
/VT200 SERIES
/WIDTH:n

Group 3: Task Setup

/[NO]ECHO
/[NO]EIGHT BIT
/[NO]ESCAPE
/[NO]FULL DUPLEX
/[NO]INTERACTIVE
/[NO]LOCAL
/[NO]PARITY[:type] !ODD is default.
 ODD
 EVEN
/[NO]PASSALL
/[NO]PASTHRU
/[NO]REMOTE
/[NO]SLAVE
/[NO]TYPEAHEAD
/[NO]WRAP

QUICK REFERENCE

SHOW ASSIGNMENTS

Section 5.11.1

SHOW ASSIGNMENTS displays at your terminal your local and login logical-device assignments. Privileged users can display assignments from other terminals and global assignments.

SHOW ASSIGNMENTS[/qualifier[s]]

Command Qualifiers

/GLOBAL	!Privileged
/LOCAL	!Default
/LOGIN	!Same as /LOCAL
/SYSTEM	!Synonym for /GLOBAL; privileged.
/TERMINAL:ttnn:	!Privileged.

SHOW CLOCK_QUEUE

Section 7.12.5

SHOW CLOCK_QUEUE displays information about tasks currently in the clock queue. This information consists of the task names, the next time each task is to run, and each task's reschedule interval, if any.

SHOW CLOCK_QUEUE

SHOW [DAY]TIME

Section 8.1.2

SHOW [DAY]TIME displays the system time and date setting.

Format

SHOW [DAY]TIME

SHOW DEFAULT

Section 4.5.2

SHOW DEFAULT displays the current default device and UFD for your terminal, along with your terminal number.

Format

SHOW DEFAULT

QUICK REFERENCE

SHOW DEVICES

Section 5.11.3

SHOW DEVICES displays information about the devices included in the system.

Format

SHOW DEVICES[/qualifier] [dd[nn:]] !Two-character mnemonic only

Command Qualifiers

/[NO]PUBLIC

/WIDTH:ddnn:

/[NO]SYSTEM

!Synonym for /PUBLIC

SHOW GROUPFLAGS

Section 6.4.2

SHOW GROUPFLAGS displays the group global event flags currently in the system.

Format

SHOW GROUPFLAGS

SHOW HOST

Section 1.2.12.2

SHOW HOST displays the name of the processor to which your terminal currently is connected. It also shows you the name and version number of the operating system running on the processor.

Format

SHOW HOST

SHOW MEMORY

Section 8.1.7

SHOW MEMORY invokes the Resource Monitoring Display (RMD), a dynamic display of the system's activities in memory.

Format

SHOW MEMORY

QUICK REFERENCE

SHOW PARTITIONS

Section 7.12.2

SHOW PARTITIONS displays address and content information about the partitions in the system. You can display information about all partitions or about a single partition.

Format

SHOW PARTITIONS[:name]

SHOW PROCESSOR

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

SHOW PROCESSOR displays information about the processors, printers, card readers, and other devices under control of the Queue Manager.

Format

SHOW processortype processorname

Processortypes:

CARD_READER	!INPUT is synonym.
DEVICE	!All nonbatch output processors; !synonym for PRINTER.
INPUT	!CARD_READER is synonym.
PRINTER	!All nonbatch output processors; !synonym for DEVICE.
PROCESSOR	

SHOW QUEUE

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

SHOW QUEUE displays information about print jobs in queues.

Format

SHOW QUEUE[/qualifier] [queuename]

Command Qualifiers

/ALL	!All entries in all queues
/BRIEF	
/DEVICE	!All nonbatch queues; synonym for /PRINTER.
/ENTRY:n	
/FILES	!Lists files in each job; not as long !as /FULL.
/FORMS:n	
/FULL	
/NAME:jobname	!Lists only jobs with that name; may !be more than one.
/OWNER_UIC:[[g,m]]	!Lists only jobs from that UIC; !default UIC is login UIC.
/PRINTER	!All nonbatch queues; synonym for !/DEVICE.

QUICK REFERENCE

SHOW SYSTEM

Section 8.1.4

SHOW SYSTEM displays information about the current system.

Format

SHOW SYSTEM[/qualifier]

Command Qualifiers

/CLI	!CLIs on current system
/DIRECTORY	!Default; displays current system UFD.
/EXTENSION LIMIT	!Task extension limit
/NETWORK UIC	!Displays current network UIC.
/PACKETS	!Maximum I/O packets and number !currently available.
/POOL	!Displays pool statistics.
/POOL/LIMITS	!Show pool limit parameters

SHOW TASKS

Section 7.12.3.1

7.12.3.2

7.12.3.3

7.12.3.4

SHOW TASKS displays information about active or installed tasks.

Format

SHOW TASKS[:taskname]/qualifier[s]

Command Qualifiers

/ACTIVE[:ttnn:]	
/DEVICE:ddnn:	!Show tasks installed for named !device.
/INSTALLED	
/LOGICAL_UNITS	!Static LUNs for installed task !qualifier.
/BRIEF	
/FULL	
/ALL	

SHOW TASKS/DYNAMIC

Section 7.12.3.4.1

7.12.3.4.2

SHOW TASKS/DYNAMIC invokes RMD to display on a video terminal continuing changes in either a single task header or of all or part of the Active Task List. On a hard-copy terminal, SHOW TASKS/DYNAMIC provides a snapshot display. See main text for more information.

QUICK REFERENCE

Format to display individual task information

SHOW TASK:taskname/DYNAMIC[/qualifier]

Command Qualifier

/RATE:n !Rate in seconds for display change; default n=1.

Format to display Active Task List

SHOW TASKS/ACTIVE/DYNAMIC[/qualifier[s]]

Command Qualifiers

/OWNER:ddnn: !Default is ALL.
 ALL

/PRIORITY:n !Highest priority to display; default n=250.

/RATE:n !Rate in seconds for display change; default n=1.

SHOW TERMINAL

Section 3.8.2

SHOW TERMINAL displays information about your terminal and other terminals on your system.

Format

SHOW TERMINAL[:ttnn:][[/qualifier] !Without qualifier, displays all
 !attributes of the terminal;
 !default ttnn:=TI:.

Command Qualifiers

/[NO]ADVANCED_VIDEO

/[NO]ANSI_CRT

/[NO]AUTOBAUD

/[NO]ASR33

/[NO]ASR35

/[NO]BLOCK_MODE

/[NO]BROADCAST

/CLI:cliname

/[NO]CRFILL

/DCL

/[NO]DEC_CRT

/DTC01

/[NO]ECHO

/[NO]EDIT_MODE

/[NO]EIGHT_BIT

/[NO]ESCAPE

/[NO]FORM_FEED

/[NO]FULL_DUPLEX

/HOSTSYNC

/[NO]HARDCOPY

/[NO]HOLD_SCREEN

/HT

!DECnet host terminal

/[NO]INTERACTIVE

/[NO]KSR33

QUICK REFERENCE

```
/[NO]LA12
/[NO]LA24
/[NO]LA30P
/[NO]LA30S
/[NO]LA34
/[NO]LA36
/[NO]LA38
/[NO]LA50
/[NO]LA100
/[NO]LA120
/[NO]LA180S
/[NO]LA210
/[NO]LFFILL
/LN03
/LQP02
/LQP03
/[NO]LOCAL
/LOGGED ON
/[NO]LOWERCASE                !NOLOWER same as /UPPER and is
                                !default.

/MCR
/MODEL
/PAGE LENGTH
/[NO]PARITY
/[NO]PASSALL
/[NO]PASTHRU
/[NO]PRIVILEGE
/PRINTER_PORT
/PRO SERIES
/[NO]REGIS
/[NO]SOFT CHARACTERS
/[NO]REMOTE
/[NO]SCOPE
/[NO]SLAVE
/SPEED
/[NO]TAB
/!TI:                          !All real terminals
/TT
/[NO]TTSYNC
/[NO]TYPE AHEAD
/[NO]UPPERCASE                !/NOUPPERCASE same as /LOWERCASE.
/[NO]VT05
/[NO]VT50
/[NO]VT52
/[NO]VT55
/[NO]VT61
/[NO]VT100
/[NO]VT101
/[NO]VT102
/[NO]VT105
/[NO]VT125
/[NO]VT131
/[NO]VT132
/[NO]VT200_SERIES
/WIDTH
/[NO]WRAP
```

SHOW USERS

Section 8.1.6

SHOW USERS displays all currently logged-in terminals, including DECnet host terminals and virtual terminals, with the default UFD and login UIC for each.

QUICK REFERENCE

Format

SHOW USERS

START

Section 7.7

START resumes execution of a task stopped by a STOP\$\$ directive.

Format

START[/qualifier] [taskname] !Taskname defaults to TTnn.

Command Qualifier

/TERMINAL:ttnn: !Privileged

START/PROCESSOR

See the RSX-11M/M-PLUS System Management Guide.

START/PROCESSOR starts a processor, batch processor, output processor, or card-reader processor. Privileged.

Format

START/processor_{type} processor_{name}[/qualifier[s]]

Processor_{types}:

APPLICATIONS_PROCESSOR	
BATCH_PROCESSOR.	
CARD_READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD_READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

Command Qualifiers

/FORMS:n	!Overrides value set on initialization.
/CONTINUE	!Default
/RESTART	
/NEXT	
/TOP OF FILE	
/BACKSPACE:n	
/FORWARDSPACE:n	
/PAGE:n	
/ALIGN	

QUICK REFERENCE

START/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

START/QUEUE starts a queue. Privileged.

Format

START/QUEUE queuename

START/QUEUE/MANAGER

See the RSX-11M/M-PLUS System Management Guide.

START/QUEUE/MANAGER starts the Queue Manager. Privileged.

Format

START/QUEUE/MANAGER

START/UNBLOCK

Section 7.7.1

START/UNBLOCK continues the execution of a task blocked by the STOP/BLOCK command. Nonprivileged users can unblock any task running from their own terminal. Privileged users can unblock any task. See main text.

Format

START/UNBLOCK [taskname]

STOP/ABORT

See the RSX-11M/M-PLUS Batch and Queue Operations Manual.

STOP/ABORT stops the current job on a line printer immediately. Privileged users can stop any job. Nonprivileged users can stop their own jobs.

Format

STOP/ABORT printer[:]

QUICK REFERENCE

STOP/BLOCK

Section 7.6

STOP/BLOCK blocks an installed running task. The task no longer executes or competes for memory. Nonprivileged users can block tasks running from their own terminals. Privileged users can block any task. See main text.

Format

STOP/BLOCK [taskname]

STOP/PROCESSOR

See the RSX-11M/M-PLUS System Management Guide.

STOP/PROCESSOR stops a processor, batch processor, card-reader processor, printer, or other output processor. Privileged.

Format

STOP/processor_{type} processor_{name}[/qualifier[s]]

Processor_{types}:

APPLICATIONS PROCESSOR	
BATCH PROCESSOR.	
CARD READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

Command Qualifiers

/ABORT	!See main text
/FILE END	
/JOB END	
/PAUSE	

STOP/QUEUE

See the RSX-11M/M-PLUS System Management Guide.

STOP/QUEUE stops queues. Privileged.

Format

STOP/QUEUE queue_{name}

QUICK REFERENCE

STOP/QUEUE/MANAGER

See the RSX-11M/M-PLUS System Management Guide.

STOP/QUEUE/MANAGER stops the Queue Manager. Privileged.

Format

STOP/QUEUE/MANAGER !Marks QMG for stop after current job.

Command Qualifier

/ABORT !Stops QMG immediately.

TYPE

Section 4.4.1

TYPE prints selected files on your terminal.

Format

TYPE [/qualifier[s]]filespec[s]

Command Qualifiers

/NOWARNINGS	!Suppresses error messages
/SHARED	!Allows other users to access file while you type it.
/DATE:dd-mmm-yy	!Given day only
/SINCE:dd-mmm-yy	!From given day through current day
/THROUGH:dd-mmm-yy	!From beginning through given day
/SINCE:dd-mmm-yy/THROUGH:dd-mm-yy	!From given day through given day
/TODAY	!Today only
/EXCLUDE:filespec	!File specification can include wildcards.

UNFIX

Section 7.11

UNFIX frees a fixed task or region from memory. Privileged.

Format

UNFIX taskname !Can be region name.

QUICK REFERENCE

UNLOCK

Section 4.4.5

UNLOCK unlocks locked files. Locked files are files that have been improperly closed. They are identified by an L in the directory listing.

Format

UNLOCK [/qualifier[s]] filespec[s]

Command Qualifiers

/NOWARNINGS	!Suppresses error messages
/DATE:dd- mm -yy	!Given day only
/SINCE:dd- mm -yy	!From given day through current day
/THROUGH:dd- mm -yy	!From beginning through given day
/SINCE:dd- mm -yy/THROUGH:dd- mm -yy	!From given day through given day
/TODAY	!Today only
/EXCLUDE:filespec	!File specification can include wildcards.

APPENDIX A

COMMON ERROR MESSAGES

The following error messages are common to many DCL commands. They may appear on your terminal preceded by a 3-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 terminal with a circumflex (^) pointing to the error. Sometimes the circumflex points to the character just past the last successfully parsed command element.

Many of the explanations refer to the Radix-50 character set. The Radix-50 characters are the uppercase alphabet, the numerals 0 through 9, the dollar sign (\$), and the period (.).

Many other error messages are included in the descriptions of the commands that produce the errors. Many system tasks also issue error messages. These messages are explained elsewhere in the documentation set. In particular, many I/O errors return error messages as code numbers. These errors are explained in the system Mini-Reference and are discussed further in the I/O documentation. The RSX-11M/M-PLUS MCR Operations Manual and the RSX-11M/M-PLUS Utilities Manual also contain explanations of many errors.

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. You may need to use the BACKUP command or the DSC utility to compress the files on the output disk.

yyy -- Allocation failure on output file

or

yyy -- Allocation failure -- no space available

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. You may need to use the BACKUP command or the DSC utility to compress the files on the output disk.

COMMON ERROR MESSAGES

yyy -- A-Z expected

Explanation: The command as typed included a nonalphabetic character.

User Action: Check command for proper syntax and reenter.

yyy -- A-Z and/or 0-9 expected

Explanation: The command as typed included a nonalphanumeric character.

User Action: Check command for proper syntax and reenter.

yyy -- Bad error message

Explanation: Some unusual condition has caused an error.

User Action: Notify your operator or system manager and make ready to send in a Software Performance Report (SPR). Record the command that caused the error and other information on activity at your terminal at the time.

yyy -- Bad use of wild cards 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 UFD not found on the current volume.

User Action: Reenter the command after checking for the correct UFD 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.

User Action: Use the COPY command to move the file from one device to another and rename it.

yyy -- Cannot truncate this filetype

Explanation: The command attempted to truncate a file that cannot be truncated. Only files containing fixed-length, variable-length, or sequenced records can be truncated.

User Action: Check to see if you have named the proper file and retry the command.

COMMON ERROR MESSAGES

- 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 by PIP.
- User Action:** Use UNLOCK to unlock the file. Determine the cause of the error and correct it if you can. You may need to check the validity of the file structure with the VFY utility.
- yyy -- Command function not unique
- Explanation:** The command as typed did not include sufficient characters to identify some command function.
- User Action:** Retype command after checking proper syntax.
- yyy -- Command line incomplete
- Explanation:** The command as typed is not a complete command.
- User Action:** Retype command after checking proper syntax.
- yyy -- Command syntax error
- Explanation:** The command did not conform to the syntax rules.
- User Action:** Check command for proper syntax and reenter it.
- yyy -- Conflicting qualifier
- Explanation:** The command as typed included qualifiers that conflict with each other in their effect.
- User Action:** Retype command after checking proper syntax.
- yyy -- Contradictory qualifier
- Explanation:** The command as typed included contradictory qualifiers, such as /DELETE and /NODELETE.
- User Action:** Check for proper syntax and reenter command.
- yyy -- Contradictory qualifier in key specification
- Explanation:** The command included a contradictory qualifier in the key definition argument to the /KEY qualifier.
- User Action:** Check command for proper syntax and reenter.
- yyy -- Decimal number expected
- Explanation:** The command included a number not in proper format.
- User Action:** Check command for proper syntax and reenter with a decimal point (.) terminating the number.

COMMON ERROR MESSAGES

yyy -- Destination expected

Explanation: You did not specify the destination parameter in the command line.

User Action: Reenter the command line after checking for proper syntax.

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 with SHOW DEVICES. Reenter command after checking for proper syntax. Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

yyy -- Device not in system

Explanation: The command specified a device that is not in the current system.

User Action: Check the devices on the system with SHOW DEVICES. Reenter command after checking for proper syntax.

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: Check the status of the device with SHOW DEVICES. Find the cause of the error and correct it if you can. It is likely that you will need to mount the volume on the device. Nonprivileged users need the volume label to mount the volume.

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 proper syntax.

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: Determine the cause of the error and correct it if you can. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

yyy -- Error dispatching command. DSW = 'n'

Explanation: An error occurred that was not explicitly handled by DCL or some invoked task.

User Action: Look up the DSW error code in the Mini-Reference or the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual. Determine the cause of the error and correct it if you can. If you cannot, see your system manager or operator.

COMMON ERROR MESSAGES

yyy -- Explicit output file name required

Explanation: The command requires an explicit output file name.

User Action: Reenter the command line in proper syntax and without wildcards.

yyy -- Extraneous input - Check for proper command structure

Explanation: The command as typed included extraneous input. The circumflex (^) points to the error or just past the last successfully parsed command element.

User Action: Reenter the command line in proper syntax.

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 line printer, failed. This error usually means the device is off-line or nonresident.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

yyy -- Failed to create output UFD

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. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

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 UFD and reenter the command. You may need privilege to delete the file.

yyy -- Failed to enter new filename

Explanation: The command specified a file that already exists in the UFD, or the directory is protected.

User Action: Check for proper syntax 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 proper file specifications and reenter the command.

COMMON ERROR MESSAGES

yyy -- Failed to get time parameters

Explanation: An internal system problem has occurred.

User Action: Retry the command. If the error recurs, see your system manager and prepare to submit a Software Performance Report. Record the command that caused the error and other information on activity at your terminal at the time.

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 UFD of the file you wish to affect. You may need to check the validity of the volume with the VFY utility.

yyy -- Failed to remove directory entry

Explanation: The command attempted to remove an entry from a directory that was either protected against access or on a write-protected device.

User Action: Determine the cause of the error and correct it if you can. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

yyy -- Failed to truncate file

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 UFD of the file you wish to affect. You may need to check the validity of the volume with the VFY utility.

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 UFD of the file you wish to affect. You may need to check the validity of the volume with the VFY utility.

yyy -- Fatal I/O error

Explanation: The command failed to execute because of some I/O error. This error can be caused by the unavailability of a device or of pool space, 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, see your operator or system manager.

COMMON ERROR MESSAGES

yyy -- File is lost

Explanation: PIP has removed a file from its UFD, failed to delete it, and failed to restore the directory entry.

User Action: Use the VFY utility to run the lost-file check to recover the file name.

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 command after checking for proper syntax.

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 -- 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 proper syntax and reenter the command.

yyy -- File specification list not available for RMS-11

Explanation: A command to an RMS-11 utility included more than one input file specification.

User Action: Check for proper syntax and reenter command.

yyy -- File version number not specified

Explanation: The command requires a file version number to be specified.

User Action: Reenter command after checking for proper syntax.

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 @ file name

Explanation: The command specified an illegal indirect command file.

User Action: Check for proper file specification and reenter the command.

COMMON ERROR MESSAGES

yyy -- Illegal command

Explanation: The command, which is the first word on the command line, is not part of DCL.

User Action: Check command for proper syntax and reenter. If the command fails again, see your system manager.

yyy -- Illegal device

Explanation: The command named a device in an illegal format or contained some other syntax error.

User Action: Check command for proper syntax and reenter. Device names are two alphabetical characters followed by an octal number and a colon.

yyy -- Illegal filespec

Explanation: The command required a file specification that was not present.

User Action: Check command for proper syntax and reenter. See Section 4.1.2 for a complete description of a file specification. Perhaps some other command element is being parsed as a filespec. Use the prompting version.

yyy -- Illegal job name

Explanation: A command to the Queue Manager named a job in an improper format.

User Action: Check command for proper syntax and reenter. Job names include an optional UIC and as many as nine Radix-50 characters. The job name is listed in the SHOW QUEUE display.

yyy -- Illegal, contradictory, or ambiguous qualifier

Explanation: One or more qualifiers to the command are in conflict, or are in error.

User Action: Check command for proper syntax and reenter.

yyy -- Illegal processor name

Explanation: A command to the Queue Manager named a processor in an improper format.

User Action: Check command for proper syntax and reenter. Processor names include as many as six Radix-50 characters. Processor names are listed in the SHOW PROCESSOR display.

yyy -- Illegal protection code

Explanation: The command specified a protection code in an improper format.

User Action: Check command for proper syntax and reenter. See Section 4.1.5 for more information on protection codes.

COMMON ERROR MESSAGES

yyy -- Illegal qualifier value

Explanation: The command as typed included an improper argument to a qualifier.

User Action: Check command for proper syntax and reenter. In DCL, an argument is preceded by a colon (:).

yyy -- Illegal queue name

Explanation: A command to the Queue Manager named a queue in an improper format.

User Action: Check command for proper syntax and reenter. Queue names include as many as six Radix-50 characters. Queue names are listed in the SHOW QUEUE display.

yyy -- Illegal task name

Explanation: The command named a task using a task name in an illegal format.

User Action: Check command for proper syntax and reenter. Task names include as many as six Radix-50 characters.

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 command for proper syntax 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 line printer or magnetic tape. Directory devices are those on which Files-11 volumes with directories can be mounted.

User Action: Correct syntax and reenter command.

yyy -- Input files have conflicting attributes

Explanation: Warning message. The command operation completed, but the files named had conflicting attributes.

User Action: Use DIRECTORY/ATTRIBUTES to find the attributes of all input and output files involved. Determine if the conflict causes any difficulty.

yyy -- Invalid command function

Explanation: The command as typed requested a function that is not valid for that command.

User Action: Check command for proper syntax 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 command for proper syntax and reenter.

COMMON ERROR MESSAGES

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 command after checking proper syntax.

yyy -- Invalid file specification qualifier

Explanation: The command included a qualifier to a file specification that was not valid. In DCL, a qualifier is preceded by a slash (/).

User Action: Check for proper syntax and retype the command.

yyy -- Invalid terminal specified

Explanation: A command directed to a specific terminal named the terminal in an improper format.

User Action: Check the name of the terminal using SHOW USERS and reenter the command with the proper format.

yyy -- Invalid time or date

Explanation: The command specified a clock or calendar field, or both, incorrectly.

User Action: Check for proper syntax and retype the command.

yyy -- Invalid UIC specified

Explanation: The command specified a 0, a number including an 8 or 9, or a number greater than 377 (octal) for the group or member number in a UIC or UFD.

User Action: Retype command after checking for proper syntax.

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 on line.
- The device is not mounted.
- 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, see your operator or system manager.

COMMON ERROR MESSAGES

yyy -- Key position size or number not specified

Explanation: The command failed to include the size or number of a key position in the key definition argument to the /KEY qualifier.

User Action: Check command for proper syntax and retype it.

yyy -- Key specification out of sequence

Explanation: The command included improper syntax in the key definition argument to the /KEY qualifier.

User Action: Check command for proper syntax and reenter.

yyy -- Magnetic tape file label not specified

Explanation: The command attempted to initialize a magnetic tape without specifying the required volume label.

User Action: Retype command after you have checked for proper syntax.

yyy -- More command parameters than permitted

Explanation: The command as typed included too many parameters.

User Action: Check command for proper syntax 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 UFD 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 a line printer.

User Action: If you can, reenter the command without specifying a UFD.

yyy -- Not logged in

Explanation: A command was entered at a terminal that was not logged in.

User Action: Only LOGIN, HELLO and HELP execute on terminals that are not logged in. If you do not know how to log in, type HELP. If you know how to log in, do so and reenter the command.

yyy -- Numeral expected

yyy -- Numeral required

Explanation: Command included nonnumeric characters in a position where numerals are required or expected.

User Action: Check command for proper syntax and reenter.

COMMON ERROR MESSAGES

yyy -- Octal number expected

Explanation: The command included a number with an 8 or 9 where an octal number was expected.

User Action: Check command for proper syntax and reenter.

yyy -- Open failure on file

Explanation: The system could not open a required file for some reason.

User Action: Check the UFD 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. If you cannot, see your operator or system manager.

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 line printer or magnetic tape. Directory devices are those on which Files-11 volumes with directories can be mounted.

User Action: Correct syntax and reenter command.

yyy -- Partition busy

Explanation: The partition into which a task was to be fixed was occupied by a running task. The requested task thus could not be fixed.

User Action: Wait until the partition is no longer busy or find out why the partition is busy and proceed accordingly.

yyy -- Primary key not specified

Explanation: A command affecting an indexed file failed to include a primary key. In many such commands, you cannot specify any action on an alternate key without first identifying the primary key.

User Action: Check for proper syntax and reenter command.

yyy -- Privilege violation

Explanation: The command was a privileged command issued at a nonprivileged terminal.

User Action: None. If you must have the function requested, find a privileged user to issue the command for you.

yyy -- Qualifier inconsistent with compiler

Explanation: A compiler command included a qualifier not acceptable by that compiler, such as a FORTRAN/F4P command including a /LINE_NUMBERS qualifier, which is for the FOR compiler only.

User Action: Check the command for proper syntax and try it again. Remember that your command may be specifying a compiler by default.

COMMON ERROR MESSAGES

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 command after checking for proper syntax.

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 commands as typed included an inappropriate argument to a qualifier. In DCL, an argument is usually preceded by a colon (:).

User Action: Check command for proper syntax and reenter.

yyy -- Radix-50 expected

Explanation: The command included a non-Radix-50 character where Radix-50 required.

User Action: Reenter the command using proper syntax. 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 command after checking for proper syntax.

yyy -- Repeated key specification

Explanation: The command as typed included the same key specification more than once in the key definition argument to the /KEY qualifier.

User Action: Retype command after checking for proper syntax.

yyy -- Repeated keyword in key specification

Explanation: The command as typed included the same keyword more than once in the key definition argument to the /KEY qualifier.

User Action: Retype command after checking for proper syntax.

yyy -- Repeated magnetic tape file label

Explanation: The command included the required volume label more than once.

User Action: Check command syntax for correctness and reenter the command.

COMMON ERROR MESSAGES

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 command after checking the syntax 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 proper syntax 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 command after checking for proper syntax.

yyy -- Required value not specified for position size or number

Explanation: The command failed to include a required value in the key definition argument to the /INDEXED qualifier.

User Action: Check command for proper syntax and reenter.

yyy -- Sorry, line too long

Explanation: DCL commands are translated for execution by MCR or another system task or utility. This error is caused by a translated command line that the destination task cannot handle.

User Action: Check command syntax 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. You may have to enter the command in MCR format or the format of the task or utility you are seeking to invoke. Check the translated command line with SET DEBUG.

yyy -- Sorry, "-" not supported in Indirect

Explanation: An indirect command file included an illegal element, as identified by the quoted part of the message.

User Action: Check indirect command file for correctness.

COMMON ERROR MESSAGES

yyy -- Sorry, low pool

Explanation: The command could not execute because of insufficient space in the system pool (dynamic storage region).

User Action: The pool is the 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.

Retry the command after an interval. If it still does not execute, see your operator or system manager. You should not attempt to execute any other task, not even an ABORT or LOGOUT, when the pool is low. These tasks too absorb pool.

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. If you cannot, see your system manager or operator. Check the task with the various SHOW TASK commands, or SHOW MEMORY.

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 command in the RUN \$ form. Determine the cause of the error and correct it if you can. If you cannot, see your system manager or operator.

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 proper syntax and reenter command.

yyy -- Version must be explicit or "*"

Explanation: The command syntax 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: Command included a wildcard (* or %) in a context where it is not permitted.

User Action: Check for proper syntax and reenter command.

yyy -- Zero value not valid for key size or number

Explanation: The command included a zero value in the key definition argument to the /KEY qualifier.

User Action: Check command for proper syntax and reenter.

APPENDIX B

FUNCTIONS INITIATED BY DCL COMMANDS

Most DCL commands initiate functions that are actually performed by MCR or by some other system task or utility. The following table gives these relationships, with any necessary comments.

DCL Command	MCR Command, Task, or Utility	Comments
ABORT	ABORT	
ALLOCATE	ALLocate	
ANALYZE/CRASH_DUMP	CDA	
ANALYZE/ERROR_LOG	ERL	
ANALYZE/MEDIA	BAD	
APPEND	PIP	
ASSIGN	ASsign	Assign logical name
ASSIGN/QUEUE	QMG	Assign queue to processor
ASSIGN/REDIRECT	REDirect	Redirect device
ASSIGN/TASK	REAssign	Alter LUN assignments
BACKUP	BRU	
BASIC	BAS, BP2	
BROADCAST	BRoadcast	
CANCEL	CANcel	
COBOL	COB	
CONTINUE	RESume	
CONVERT	RMS CNV	
COPY	PIP	
CORAL	COR	
CREATE	PIP, FCS	
CREATE/DIRECTORY	UFD	

FUNCTIONS INITIATED BY DCL COMMANDS

DCL Command	MCR Command, Task, or Utility	Comments
DEALLOCATE	DEAllocate	
DEASSIGN	ASsign	Deassign logical name
DEASSIGN/QUEUE	QMG	Eliminate assignment of queue to processor
DELETE	PIP	Delete files
DELETE/ENTRY	QMG	
DELETE/JOB	QMG	
DELETE/PROCESSOR	QMG	Delete despooler task
DELETE/QUEUE	QMG	Delete queue
DIFFERENCES	CMP	
DIRECTORY	PIP RMS DSP	/ATTRIBUTES qualifier
DISMOUNT	DisMOunt	
EDIT	EDT, EDI, KED, SLP	
FORTRAN	FOR, F4P, F77	
FIX	FIX	
HELP	HELlo	As part of HELlo, HELP can run on terminals that are not logged in.
HOLD/ENTRY	QMG	
HOLD/JOB	QMG	
INITIALIZE	INItvolume	Create Volume
INITIALIZE/PROCESSOR	QMG	Set up despooler task
INITIALIZE/QUEUE	QMG	
INITIALIZE/UPDATE	HOME	Alter volume's home block
INSTALL	INStall	
LIBRARY	LBR	
LINK	TKB	Task Builder. May be more than one on system. FTB is Fast TKB.
LOGIN	HELlo	
LOGOUT	BYE	

FUNCTIONS INITIATED BY DCL COMMANDS

DCL Command	MCR Command, Task, or Utility	Comments
MACRO	MAC	
MCR	MCR	
MOUNT	MOUnt	
PRINT	QMG	
PURGE	PIP	
RELEASE/ENTRY	QMG	
RELEASE/JOB	QMG	
REMOVE	REMOve	
RENAME	PIP	
REQUEST	BRoadcast	
RUN	RUN INS	
SET [DAY]TIME	TIME	
SET DEFAULT	SET /UIC	SET for UFD ASN for SY:
SET DEVICE	DEVIce	
SET FILE	PIP	
SET GROUPFLAGS	FLAg	
SET HOST	SET /HOST	
SET PARTITION	PAR, SET /MAIN, SET /PAR, SET SUB	
SET PRIORITY	ALTer	
SET PROTECTION	PIP	
SET QUEUE	QMG	
SET SYSTEM/DIRECTORY	SET /SYSUIC	
SET SYSTEM/EXTENSION_LIMIT	SET /MAXEXT	
SET SYSTEM/NETWORK_UIC	SET /NETUIC	
SET SYSTEM/NOLOGINS	SET /NOLOGON	
SET SYSTEM/PACKETS	SET /MAXPKT	
SET SYSTEM/POOL	SET /POOL	
SET SYSTEM/POOL/LIMITS	SET /PLCTL	
SET TERMINAL	SET /attribute	
SHOW ASSIGNMENTS	ASsigN	

FUNCTIONS INITIATED BY DCL COMMANDS

DCL Command	MCR Command, Task, or Utility	Comments
SHOW CLOCK_QUEUE	CLQ	
SHOW [DAY]TIME	TIM	
SHOW DEFAULT	SET /UIC ASSIGN	SET for UFD ASN for SY:
SHOW DEVICES	DEV SET /PUB SET /BUF	/PUBLIC qualifier /WIDTH qualifier
SHOW GROUPFLAGS	FLAg	
SHOW HOST	SET /HOST	
SHOW MEMORY	RMD	Resource Monitoring Display
SHOW PARTITIONS	PAR	
SHOW PROCESSOR	QMG	
SHOW QUEUE	QMG	
SHOW SYSTEM/CLI	CLI /SHOW	
SHOW SYSTEM/DIRECTORY	SET /SYSUIC	
SHOW SYSTEM/EXTENSION_LIMIT	SET /MAXEXT	
SHOW SYSTEM/NETWORK_UIC	SET /NETUIC	
SHOW SYSTEM/POOL	SET /POOL	
SHOW SYSTEM/POOL/LIMITS	SET /PLCTL	
SHOW TASKS/ACTIVE	ACT	
SHOW TASK:taskname/ACTIVE	ATL	
SHOW TASKS/DYNAMIC	RMD	
SHOW TASKS/INSTALLED	TAS	
SHOW TASKS/INSTALLED/FULL	TAL	
SHOW TASK:taskname/INSTALLED	TAS	
SHOW TERMINAL	DEV	
SHOW TERMINAL/attribute	SET /attribute	
SHOW USERS	DEV	
SORT	SRT	SORT-11
START	UNS	
START/PROCESSOR	QMG	

FUNCTIONS INITIATED BY DCL COMMANDS

DCL Command	MCR Command, Task, or Utility	Comments
START/QUEUE	QMG	
START/QUEUE/MANAGER	QMG	
START/UNBLOCK	UNBLock	
STOP/ABORT	QMG	
STOP/BLOCK	Block	
STOP/PROCESSOR	QMG	
STOP/QUEUE	QMG	
STOP/QUEUE/MANAGER	QMG	
TYPE	PIP	
UNFIX	UNFix	
UNLOCK	PIP	

INDEX

- Abbreviation
 - DCL, 1-7
- ABORT command, 7-16 to 7-20, 9-2
- Access
 - types, 4-9
- Account file, 3-9
- Account File Maintenance Program
 - See ACNT
- ACNT, 3-3, 3-4, 3-9
- ACP, 5-3
 - DEctape, 5-39
 - disk, 5-39
 - foreign, 5-39
 - magnetic tape, 5-39
 - specifying, 5-39
- ACT command
 - MCR, 3-2
- Active Task List
 - See ATL
- ALLOCATE command, 5-31 to 5-33, 9-2
- ALTMODE key, 3-10
- Ampersand (&), 3-14
- ANALYZE/CRASH_DUMP command, 9-2
- ANALYZE/MEDIA command, 5-14 to 5-20, 9-3
- Ancillary Control Processor
 - See ACP
- Angle bracket
 - left (<), 3-13
 - right (>), 3-13
 - CLI prompt, 3-13
- ANSI, 5-3
 - file name, 1-9
 - magnetic tape
 - preparing, 5-12
- APPEND command, 4-70 to 4-71, 9-4
- Arrow key, 3-15
- ASSIGN command, 5-21 to 5-23, 9-4
- ASSIGN/QUEUE command, 9-5
- ASSIGN/REDIRECT command, 5-24 to 5-26, 9-5
- ASSIGN/TASK command, 5-27 to 5-28, 9-5
- Assignment
 - device
 - global, 5-8
 - local, 5-8
 - login, 5-8
 - logical-device
 - displaying, 2-5, 5-86
- Asterisk (*), 3-14, 4-6 to 4-8
- At sign (@), 3-14
- ATL
 - displaying task information, 7-57
 - task
 - displaying information, 7-59
- BACK SPACE key, 3-11
- Backslash (\), 3-15
- Backup and Restore Utility
 - See BRU utility
- BACKUP command, 5-73 to 5-85, 9-5 to 9-7
 - error messages, 5-83 to 5-85
 - qualifiers, 5-75 to 5-82
- Backup operation, 5-73
- Backup set, 5-73
- Bad block
 - locating, 5-14
- BADBLK.SYS file, 5-11
- Base number
 - in DCL, 1-8
- BITMAP.SYS file, 5-11
- Brace ({ }), 3-15
- BROADCAST command, 3-28 to 3-31, 9-7
 - error messages, 3-30
 - examples, 3-29
 - qualifiers, 3-29
- BRU utility, 5-83
- CANCEL command, 7-23, 9-7
- CAPS LOCK key, 3-12
- CBD
 - entry
 - removing, 7-37
- Checkpoint file, 5-88
- Circumflex (^), 3-13, 3-16
- CL:, 5-5
- CLI, 1-1
 - changing, 3-3
 - displaying information, 8-6
- Clock queue, 7-11
 - entry
 - displaying information, 7-65
 - eliminating, 7-23
 - inserting, 7-23
- CMP utility, 4-77
- CO:, 5-5

INDEX

- Colon (:), 3-14
 - DCL, 1-9
- Command
 - aborting, 7-16
 - DCL
 - abbreviation, 9-1
 - function, 1-15 to 1-17
 - translating to MCR, 1-13 to 1-14, 2-8, B-1
- Command line
 - continuing, 3-10
 - DCL
 - comment, 1-12
 - quoting string, 1-9
 - terminating, 3-10
- Command line interpreter
 - See CLI
- Comment
 - command line
 - DCL, 1-12
- Common
 - shareable
 - building, 6-20
- Common Block Directory
 - See CBD
- Compiler, 6-2
- Console Logging, 5-5
- Continuation line
 - See Line continuation
- CONTINUE command, 7-21 to 7-22, 9-7
- Control key
 - See CTRL key
- Controller, 5-1
- CONVERT command, 4-50 to 4-56, 9-8
- COPY command, 4-42 to 4-46, 9-8
- COPY key, 3-12
- Copying
 - file, 4-42
- CORIMG.SYS file, 5-11
- Create access, 4-10
- CREATE command, 4-10, 4-11 to 4-12, 9-9
- CREATE/DIRECTORY command, 4-13 to 4-15, 9-9
 - examples, 4-14
 - qualifiers, 4-13 to 4-14
- CTRL key, 3-16
- CTRL/C, 3-10, 3-17
- CTRL/G, 3-20
- CTRL/I, 3-20
- CTRL/K, 3-20
- CTRL/L, 3-20
- CTRL/M, 3-20
- CTRL/O, 3-16, 3-18
- CTRL/Q, 3-17
- CTRL/R, 3-19
- CTRL/S, 3-17
- CTRL/U, 3-19
- CTRL/X, 3-20
- CTRL/Z, 3-10, 3-16, 3-18
- Cursor
 - controlling, 3-15
- Cursor-control key, 3-15

- Date
 - related qualifiers, 4-8
 - system
 - displaying, 2-1, 8-3
 - setting, 2-1, 8-1
- Date format
 - DCL, 1-8
- DCB, 5-5
- DCL, 1-1
 - abbreviations, 1-7
 - colon (:), 1-9
 - command
 - abbreviation, 9-1
 - function, 1-15 to 1-17
 - translating to MCR, 2-8, B-1
 - command line, 1-2
 - quoting string, 1-9
 - comment, 1-12
 - dates, 1-8
 - DECnet, 1-10
 - equal sign (=), 1-9
 - error messages
 - common, A-1
 - errors, 1-14
 - line continuation, 1-12, 3-10
 - number
 - decimal, 1-8
 - octal, 1-8
 - parameter, 1-8
 - qualifier, 1-3 to 1-4
 - quotation mark (")
 - using, 1-9
 - RSX-11M, 1-19
 - SET commands, 2-1
 - SHOW commands, 2-1
 - underscore (_), 1-8
 - VAX/VMS, 1-19
- DCL command, 1-6
- DEALLOCATE command, 5-34 to 5-35, 9-9
- DEASSIGN command, 5-29 to 5-30, 9-10
- DEASSIGN/QUEUE command, 9-10

INDEX

- DECnet
 - DCL, 1-10
 - directory
 - displaying, 8-6
 - setting, 8-4
- DECTape, 5-3
 - ACP, 5-39
- DECwriter II, 3-1
- Delete access, 4-10, 4-84
- DELETE command, 4-35 to 4-38, 9-10
- DELETE key, 3-11
- DELETE/ENTRY command, 9-11
- DELETE/JOB command, 9-11
- DELETE/processor type command, 9-11
- DELETE/QUEUE command, 9-12
- Density
 - volume
 - changing, 5-70
 - setting, 5-59
- Device, 5-1, 5-4, 5-5
 - allocating, 5-31
 - attribute
 - displaying, 5-91
 - setting, 5-88
 - buffer
 - setting, 5-89
 - deallocating, 5-34
 - default, 4-2, 4-4
 - displaying, 4-82
 - setting, 4-80
 - displaying information, 2-3, 5-91
 - file
 - default number, 5-62
 - global assignment, 5-8
 - industrial, 5-4
 - laboratory, 5-4
 - list, 5-2
 - local assignment, 5-8
 - logical assignment
 - displaying, 2-5, 5-86
 - logical name, 5-5, 5-6, 5-8
 - assigning, 5-8, 5-21
 - deleting, 5-29
 - table, 5-8
 - login assignment, 5-8
 - mass-storage, 5-3
 - name, 5-1
 - owner, 5-9
 - peripheral, 5-1
 - physical, 5-4
 - private, 5-9, 5-12
 - allocating, 5-31
- Device
 - private (Cont.)
 - deallocating, 5-34
 - protection, 5-12
 - pseudo, 5-1, 5-4, 5-5, 5-5 to 5-6
 - list, 5-6
 - name, 5-5
 - public, 5-9, 5-13
 - setting, 5-89
 - random-access, 5-3
 - redirecting, 5-5
 - output, 5-24
 - sequential-access, 5-3
 - setting default, 2-5
 - shareable, 5-9, 5-12
 - unit record, 5-3
 - unowned, 5-9, 5-13
- Device Control Block
 - See DCB
- Device driver, 5-1
 - name, 5-4
- DIFFERENCES command, 4-72 to 4-77, 9-12
- DIGITAL Command Language
 - See DCL
- DIGITAL Standard Editor
 - See EDT
- Directory, 4-1
 - creating, 4-10, 4-13
- DECnet
 - displaying, 8-6
 - setting, 8-4
 - default, 4-2, 4-4
 - displaying, 4-82
 - setting, 4-80
 - displaying information, 4-26
 - setting default, 2-5
 - system, 7-6, 8-7
 - displaying, 2-8, 8-6
 - setting, 2-8, 8-3, 8-7
- DIRECTORY command, 4-26 to 4-34, 9-13
 - examples, 4-29 to 4-33
 - qualifiers, 4-27 to 4-29
- Disk, 5-3
 - ACP, 5-39
 - bad blocks
 - locating, 5-14
 - preparing, 5-9
- DISMOUNT command, 5-53 to 5-56, 9-13
- Dollar sign (\$)
 - DCL prompt, 3-13

INDEX

- Driver
 - device, 5-1
- Dynamic storage region
 - See Pool
- Echo, 3-9
- EDI, 4-23
- EDIT command, 4-15, 4-16, 9-14
- EDIT/EDI command, 4-23, 9-14
- EDIT/editor command, 4-23, 9-14
 - K52, 4-24
 - KED, 4-24
 - SOS, 4-23
 - TECO, 4-23
- EDIT/EDT command, 4-16, 9-14
- EDIT/SLP command, 4-20, 9-14
- Editing
 - file, 4-16, 4-23
- Editor
 - EDI, 4-23
 - EDT, 4-16
 - K52, 4-24
 - KED, 4-24
 - SOS, 4-23
 - TECO, 4-23
- EDT, 4-16
- ENTER key, 3-15
- Entry point name, 6-28
 - removing, 6-43
- Entry point table
 - See EPT
- EPT, 6-28
- Equal sign (=), 3-14
 - DCL, 1-9
- Error
 - DCL, 1-14
- Error messages
 - DCL
 - common, A-1
- ESC key, 3-10
- ESCAPE key, 3-10
- Event flag
 - group global, 6-51
 - creating, 6-52
 - deleting, 6-52
 - displaying, 6-53
- Extend access, 4-10, 4-84
- Extension
 - file
 - See File type
- FillACP, 5-3, 5-8
- Fast Task Builder
 - See FTB
- File, 4-1, 5-1
 - access, 2-7
 - access types, 4-9, 4-84
 - attribute
 - setting, 4-87
 - checkpoint, 5-88
 - comparing, 4-72
 - copying, 4-42
 - default number, 5-62
 - deleting, 4-35
 - displaying, 4-57
 - displaying information, 4-26
 - editing, 4-16, 4-23
 - excluding, 4-9
 - library, 6-28
 - locked, 4-78
 - maximum number
 - changing, 5-71
 - setting, 5-62
 - object
 - binary, 6-2
 - printing, 4-60
 - protection, 4-9
 - code, 4-9, 4-83
 - default, 4-84
 - setting, 4-83, 5-38, 5-63, 5-64, 5-71
 - purging, 4-39
 - renaming, 4-47
 - sequential
 - appending, 4-70
 - creating, 4-10, 4-11
 - source
 - maintaining, 4-20
 - system, 5-10
 - system library, 6-2
 - task image
 - name
 - specifying, 7-28
 - TKB map, 6-2
 - producing, 6-13
 - unlocking, 4-78
 - version number, 4-3
 - changing, 4-47
 - default, 4-4
- File Compare Utility
 - See CMP utility
- File header, 4-2
 - allocating, 5-60
 - maximum number, 5-60
- File name, 4-3
 - ANSI, 1-9
 - changing, 4-47
- File protection
 - setting, 2-6

INDEX

- File record
 - moving, 4-50
- File Set ID, 5-38, 5-48
- File Set Identifier
 - See File Set ID, 5-38
- File specification, 4-2
 - defaults, 4-4
 - wildcards, 4-6
- File type, 4-3
 - changing, 4-47
 - default, 4-4
 - standard, 4-5
- Files-11
 - volume, 5-3
 - backing up, 5-73
 - formatting, 5-57
 - level 401, 5-62
 - level 402, 5-62
 - restoring, 5-73
- FIX command, 7-38 to 7-39, 9-15
- Floating qualifier, 1-3 to 1-4
- Form feed, 3-20
- FTB, 6-11

- GEN partition, 7-2
- Global assignment
 - device, 5-8
- Global symbol, 6-28
 - removing, 6-43
- Group global event flag, 6-51
 - creating, 6-52
 - deleting, 6-52
 - displaying, 6-53

- HELP command, 1-4 to 1-6, 3-21 to 3-24, 9-15
 - examples, 3-23 to 3-24
 - qualifiers, 3-22
- Help file
 - example, 3-25 to 3-27
 - format, 3-24 to 3-27
 - writing, 3-24 to 3-27
- HOLD qualifier
 - LOGOUT command, 3-8
- HOLD/ENTRY command, 9-16
- HOLD/JOB command, 9-16
- Home block
 - volume, 5-57
 - changing, 5-70
 - displaying information, 5-71
- HOME command, 5-70
- HOME utility
 - See HOME command
- Host system
 - displaying, 1-11 to 1-12

- Host system (Cont.)
 - setting, 1-10 to 1-11
- Hyphen (-), 3-14

- Index file
 - locating, 5-61
- INDEXF.SYS file, 5-10
- Indirect, 1-17
- Indirect command file
 - DCL, 1-17
- Indirect Command Processor
 - See Indirect
- INITIALIZE command, 5-57 to 5-69, 9-16
 - error messages, 5-66 to 5-69
 - qualifiers, 5-58 to 5-65
- INITIALIZE/PROCESSOR command
 - input, 9-17
 - output, 9-17
- INITIALIZE/QUEUE command, 9-18
- INITIALIZE/UPDATE command, 5-70 to 5-72, 9-18
 - qualifiers, 5-70 to 5-71
- INSTALL command, 7-28 to 7-36, 9-19
 - error messages, 7-31 to 7-36
 - qualifiers, 7-28 to 7-30
- Install-run-remove form
 - RUN command, 7-3, 7-7

- Journal file, 4-17

- Key
 - ALTMODE, 3-10
 - arrow, 3-15
 - BACK SPACE, 3-11
 - CAPS LOCK, 3-12
 - COPY, 3-12
 - CTRL, 3-16
 - cursor-control, 3-15
 - DELETE, 3-11
 - ENTER, 3-15
 - ESC, 3-10
 - ESCAPE, 3-10
 - LINE FEED, 3-11
 - NO SCROLL, 3-12
 - NUMBERS, 3-15
 - PERIOD, 3-15
 - PF, 3-15
 - REPEAT, 3-12
 - repeating, 3-12
 - RETURN, 3-2, 3-10
 - RUBOUT, 3-11
 - SCROLL, 3-12
 - SEL, 3-10

INDEX

- Key (Cont.)
 - SHIFT, 3-12
 - TAB, 3-20
- Keyboard
 - terminal, 3-1, 3-9
- Keypad
 - terminal, 3-1, 3-15
 - VT200-series, 3-15
- LA36, 3-1
- Label
 - volume, 5-37, 5-58
 - changing, 5-71
- Language
 - compiler, 6-2
- LB:, 5-5, 7-6, 8-7
- LBR, 6-29
- Left-angle bracket (<), 3-13
- Librarian utility
 - See LBR
- Library, 6-28
 - creating, 6-28, 6-32
 - file, 6-28
 - system, 6-2
 - macro, 6-28
 - maintaining, 6-28
 - module, 6-28
 - deleting, 6-35
 - extracting, 6-36
 - inserting, 6-32, 6-37
 - listing, 6-39
 - replacing, 6-44
 - system, 5-5
- LIBRARY command, 6-28 to 6-31, 9-19
 - error messages, 6-45 to 6-51
 - functions, 6-28
- LIBRARY/COMPRESS command, 6-31 to 6-32, 9-19
- LIBRARY/CREATE command, 6-32 to 6-34, 9-20
- LIBRARY/DELETE command, 6-35 to 6-36, 9-20
- LIBRARY/EXTRACT command, 6-36 to 6-37, 9-20
- LIBRARY/INSERT command, 6-37 to 6-39, 9-21
- LIBRARY/LIST command, 6-39 to 6-43, 9-21
- LIBRARY/REMOVE command, 6-43, 9-21
- LIBRARY/REPLACE command, 6-44 to 6-45, 9-22
- Line continuation
 - DCL, 1-12, 3-10
- LINE FEED key, 3-11
- Line printer, 5-3
 - lowercase, 5-89
- Line Text Editor
 - See EDI
- LINK command, 6-2, 6-7 to 6-27, 9-22 to 9-23
 - examples, 6-24 to 6-26
 - functions, 6-2
 - /OPTION qualifier, 6-4
 - process, 6-4
 - qualifiers, 6-8 to 6-24
 - /SAVE qualifier, 6-5
- Linking tasks, 6-2
- Local assignment
 - device, 5-8
- Logging in, 3-2
- Logging out, 3-2
- Logical device
 - assignment
 - displaying, 2-5, 5-86
- Logical name, 5-5
 - device, 5-5, 5-6, 5-8
 - assigning, 5-8, 5-21
 - deleting, 5-29
 - table, 5-8
- Logical unit number
 - See LUN
- Logical Unit Table
 - See LUT
- Login
 - disabling, 8-4
 - enabling, 8-4
- Login assignment
 - device, 5-8
- LOGIN command, 3-4 to 3-7, 9-23
 - error messages, 3-7
 - examples, 3-5 to 3-6
- LOGIN.CMD file, 3-4
- LOGIN.TXT file, 3-4
- LOGOUT command, 3-8 to 3-9, 5-13, 9-23
 - examples, 3-8
 - HOLD qualifier, 3-8
- Lowercase, 5-89
- LUN, 5-5, 5-7
 - assigning, 5-7
 - assignment
 - default, 5-7
 - displaying, 7-61
 - reassigning, 5-27
- LUT, 5-5
- Macro library, 6-28

INDEX

- MACRO-11 Relocatable Assembler, 6-2
- Magnetic tape, 5-3
 - ACP, 5-39
 - ANSI
 - preparing, 5-12
 - mounting, 5-48
 - RMS-11, 5-44
 - set, 5-48
 - mounting, 5-48
- Map file
 - TKB, 6-2
 - producing, 6-13
- Master File Directory
 - See MFD
- MCR, 1-1
- MCR command, 1-6, 9-24
- Memory, 7-2
 - displaying, 2-2, 8-9
- Message
 - sending, 3-28, 3-32
- MFD, 4-1, 5-10
- MNT, 6-28
- Module name table
 - See MNT
- Monitor Console Routine
 - See MCR
- MOUNT command, 5-8, 5-36 to 5-47, 9-24 to 9-25
 - error messages, 5-49 to 5-52
 - examples, 5-46
 - qualifiers, 5-38 to 5-45
- MOUNT/FOREIGN command, 5-8
- MTAAC, 5-3
- MTAACP, 5-48

- NL:, 5-6
- NO SCROLL key, 3-12
- Node name
 - specifying, 1-10
- Null device, 5-6
- Number sign (), 3-14
- NUMBERS key, 3-15

- Object file
 - binary, 6-2
- Object module, 6-2
 - deleting, 6-31
 - library, 6-28
 - deleting, 6-35
 - extracting, 6-36
 - inserting, 6-32, 6-37
 - listing, 6-39
 - replacing, 6-44
 - linking, 6-7
- Object module (Cont.)
 - name, 6-28

- Parameter
 - DCL, 1-8
- Parenthesis (()), 3-15
- Partition, 7-2, 7-41
 - base, 7-42
 - boundary
 - changing, 7-42
 - creating, 7-41
 - eliminating, 7-41
 - information
 - displaying, 7-46
 - size, 7-42
 - system-controlled, 7-42
 - types, 7-42
- Password, 3-5
- Percent sign (%), 3-14, 4-8
- Period (.), 3-13
- PERIOD key, 3-15
- Peripheral device, 5-1
- Peripheral Interchange Program
 - See PIP utility
- PF key, 3-15
- PIP utility, 1-18
- Plus sign (+), 3-14
- PMD
 - specifying, 7-16, 7-29
- PMT
 - parameters
 - displaying, 8-7
 - setting, 8-4
- Pool, 7-1
 - displaying contents, 8-9
 - displaying information, 8-7
 - limits
 - setting, 8-4
 - size
 - increasing, 8-4
- Postmortem Dump
 - specifying, 6-15
- PRINT command, 4-60 to 4-69, 9-25
 - examples, 4-66 to 4-68
 - qualifiers, 4-60 to 4-66
- Print job, 4-60
- Priority
 - task, 7-2
- Program
 - developing, 6-1
 - source, 6-1
- Prompt, 1-3
 - explicit, 3-17

INDEX

- Pseudo device, 5-1, 5-4, 5-5, 5-5
 - to 5-6
 - list, 5-6
 - name, 5-5
- PURGE command, 4-39 to 4-41, 9-26
- QMG, 1-17, 2-4, 4-60
 - displaying information
 - devices, 2-4
 - queue file, 2-4
- Qualifier
 - date-related, 4-8
 - DCL, 1-3 to 1-4
 - floating, 1-3 to 1-4
- Queue file
 - displaying information, 2-4
- Queue Manager
 - See QMG
- Quotation mark ("), 1-9
- Radix-50, A-1
- Read access, 4-9, 4-84
- Record Management Services
 - See RMS-11
- Redirection, 5-5, 5-24
- Region
 - fixing, 7-38
 - removing, 7-37
- RELEASE/ENTRY command, 9-26
- RELEASE/JOB command, 9-26
- REMOVE command, 7-37, 9-26
- REMOVE/REGION command, 7-37
- RENAME command, 4-47 to 4-49, 9-27
- REPEAT key, 3-12
- REQUEST command, 3-32, 9-27
- Resource Monitoring Display
 - See RMD
- RETURN key, 3-2, 3-10
- Right-angle bracket (>), 3-13
 - CLI prompt, 3-13
- RMD, 8-9
- RMS-11, 4-50
 - magnetic tape, 5-44
- RMSCNV utility, 4-50
- RUBOUT key, 3-11
- RUN command, 7-5, 7-11 to 7-15, 9-27, 9-28
 - install-run-remove form, 7-3, 7-7 to 7-11
- SCROLL key, 3-12
- Scrolling
 - terminal, 3-12
- SEL key, 3-10
- Sequential file
 - appending, 4-70
 - creating, 4-10, 4-11
- SET commands
 - DCL, 2-1
 - SET DAYTIME command, 2-1, 8-1, 9-28
 - SET DEBUG command, 1-2, 1-13 to 1-14, 2-8, 9-29
 - SET DEFAULT command, 2-5, 4-80 to 4-81, 9-29
 - SET DEVICE command, 5-88 to 5-90, 9-29
 - SET FILE command, 4-87 to 4-89, 9-29
 - SET GROUPFLAGS command, 6-52, 9-30
 - SET HOST command, 1-10 to 1-11, 1-11 to 1-12, 9-30
 - SET NODEBUG command, 1-13 to 1-14, 2-8
 - SET NOPARTITION command, 9-30
 - SET PARTITION command, 7-41 to 7-45, 9-30
 - SET PRIORITY command, 7-63 to 7-64, 9-31
 - SET PROTECTION command, 2-6, 4-83 to 4-86, 9-31
 - SET QUEUE/ENTRY command, 9-31
 - SET QUEUE/JOB command, 9-32
 - SET SYSTEM command, 8-3 to 8-5, 9-32
 - qualifiers, 8-3 to 8-5
 - SET SYSTEM/DIRECTORY command, 2-8, 8-7
 - SET TERMINAL command, 2-3, 3-34 to 3-46, 9-33 to 9-34
 - examples, 3-45
 - qualifiers, 3-35 to 3-45
 - SET TIME command, 2-1, 8-1, 9-28
 - SHIFT key, 3-12
 - SHOW /CLOCK_QUEUE command, 7-65
 - SHOW ASSIGNMENTS command, 2-5, 5-86 to 5-87, 9-35
 - SHOW CLOCK_QUEUE command, 9-35
 - SHOW commands
 - DCL, 2-1
 - SHOW DAYTIME command, 2-2, 8-3, 9-35
 - SHOW DEFAULT command, 2-5, 4-82, 9-35
 - SHOW DEFAULTS command, 3-2
 - SHOW DEVICES command, 2-3, 3-3, 5-91 to 5-94, 9-36

INDEX

- SHOW GROUPFLAGS command, 6-53, 9-36
- SHOW HOST command, 9-36
- SHOW MEMORY command, 2-2, 8-9 to 8-12, 9-36
- SHOW PARTITIONS command, 7-46 to 7-48, 9-37
- SHOW PROCESSOR command, 2-4, 9-37
- SHOW QUEUE command, 2-4, 9-37
- SHOW QUEUE/FULL command, 2-4
- SHOW SYSTEM command, 8-6 to 8-7, 9-38
 - qualifiers, 8-6 to 8-7
- SHOW SYSTEM/DIRECTORY command, 2-8, 8-7
- SHOW TASK/LOGICAL UNITS command, 7-61 to 7-62
- SHOW TASK:taskname/DYNAMIC command, 7-57 to 7-59, 9-39
- SHOW TASKS command, 9-38
- SHOW TASKS command, 2-5, 7-49 to 7-62
- SHOW TASKS/ACTIVE command, 3-2, 7-2, 7-49 to 7-54
- SHOW TASKS/ACTIVE/DYNAMIC command, 7-59 to 7-61, 9-39
- SHOW TASKS/DYNAMIC command, 9-38
- SHOW TASKS/DYNAMIC command, 7-57
- SHOW TASKS/INSTALLED command, 2-6, 7-54 to 7-56
- SHOW TASKS/INSTALLED/DEVICE command, 7-56 to 7-57
- SHOW TERMINAL command, 2-2, 3-47 to 3-57, 9-39 to 9-40
 - examples, 3-51 to 3-57
 - qualifiers, 3-48 to 3-50
- SHOW TIME command, 2-2, 8-3, 9-35
- SHOW USERS command, 2-6, 8-8, 9-40
- SHOW/TASKS/ACTIVE command, 2-5
- Slash (/), 3-15
- SLP utility, 4-20, 4-74
- Source file
 - maintaining, 4-20
- Source Language Input Program
 - See SLP utility
- Square bracket ([]), 3-15
- START command, 7-26, 9-41
- START/PROCESSOR command, 9-41
- START/QUEUE command, 9-42
- START/QUEUE/MANAGER command, 9-42
- START/UNBLOCK command, 7-27, 9-42
- STD, 7-1, 7-2
 - entry
 - including, 7-28
- STD
 - entry (Cont.)
 - removing, 7-37
- STOP/ABORT command, 9-42
- STOP/BLOCK command, 7-24 to 7-25, 9-43
- STOP/PROCESSOR command, 9-43
- STOP/QUEUE command, 9-43
- STOP/QUEUE/MANAGER command, 9-44
- String
 - quoting, 1-9
- Subpartition, 7-41
- SY:, 4-2, 4-4, 5-6
- SYSLOGIN.CMD file, 3-4
- System
 - attribute
 - displaying, 8-6
 - setting, 8-3
 - directory, 8-7
 - displaying, 8-6, 8-7
 - setting, 8-3, 8-7
 - file, 5-10
 - host
 - displaying, 1-11 to 1-12
 - setting, 1-10 to 1-11
 - multiuser protection, 3-3
 - pool
 - displaying contents, 8-9
 - displaying information, 8-7
 - increasing, 8-4
 - setting limits, 8-4
- System date
 - displaying, 2-1
 - setting, 2-1
- System directory, 7-6
 - displaying, 2-8
 - setting, 2-8
- System library, 5-5
- System Task Directory
 - See STD
- System time
 - displaying, 2-1
 - setting, 2-1
- Tab
 - vertical, 3-20
- TAB key, 3-20
- Task, 7-1
 - aborting, 7-16
 - active, 7-1
 - displaying information, 7-2, 7-49
 - blocking, 7-24
 - building, 6-2
 - checkpointable, 7-2

INDEX

- Task
 - checkpointable (Cont.)
 - specifying, 7-28
 - creating, 6-7
 - displaying information, 2-5, 7-57
 - dormant, 7-1
 - executing, 7-5
 - information
 - displaying, 7-57
 - installed, 7-1
 - displaying information, 7-49, 7-54, 7-56
 - dormant, 7-28
 - executing, 7-6, 7-11
 - fixing, 7-38
 - LUNs
 - displaying, 7-61
 - removing, 7-7
 - unfixing, 7-40
 - installing, 7-1, 7-7, 7-28
 - linking, 6-2
 - name, 7-2
 - default, 7-3
 - removing from STD, 7-37
 - specifying, 7-30
 - priority, 7-2
 - changing, 7-63
 - specifying, 7-29
 - removing, 7-37
 - resuming, 7-21, 7-26, 7-27
 - scheduling, 7-7, 7-11
 - size
 - displaying, 8-6
 - extending, 8-3
 - status flags, 7-51 to 7-52
 - stopping (?), 7-24
 - UIC
 - specifying, 7-30
 - uninstalled
 - executing, 7-7
 - Task build
 - aborting, 6-3, 6-11
 - Task Builder
 - See TKB
 - Task Control Block
 - See TCB
 - Task image, 6-2
 - file, 7-5, 7-7
 - specifying name, 7-28
 - Task Termination Notification
 - program
 - TCB, 7-1
 - Terminal
 - attached, 3-2
- Terminal (Cont.)
 - attribute
 - displaying, 2-2
 - setting, 2-2
 - character
 - deleting, 3-11, 3-19
 - characteristic
 - displaying, 2-2, 3-47
 - setting, 2-2, 3-34
 - displaying information, 3-2, 3-47
 - lowercase, 5-89
 - model, 2-3
 - displaying, 2-2
 - setting, 2-2
 - scrolling, 3-12
 - slaved, 3-2
 - typeahead buffer
 - clearing, 3-20
 - Terminal characteristics
 - displaying, 3-2
 - setting, 3-2
 - Terminal keyboard, 3-1, 3-9
 - Terminal keypad, 3-1, 3-15
 - VT200-series, 3-15
 - Terminal number, 3-1
 - Terminal output
 - interrupting, 3-17
 - skipping, 3-18
 - TI:, 5-6
 - Time
 - system
 - displaying, 2-1, 8-3
 - setting, 2-1, 8-1
 - TKB, 6-2
 - examples, 6-5
 - format, 6-5
 - invoking, 6-2, 6-7
 - map file, 6-2
 - producing, 6-13
 - options, 6-2, 6-4, 6-14
 - list, 6-16 to 6-19
 - TKTN, 7-19
 - messages, 7-19 to 7-20
 - TYPE command, 4-57 to 4-59, 9-44
 - Typeahead buffer
 - clearing, 3-20
 - UCB, 5-5
 - UFD, 4-1
 - creating, 4-10, 4-13
 - UIC, 3-4, 4-2
 - task
 - specifying, 7-30

INDEX

- Underscore (), 3-14
 - DCL, 1-8
- UNFIX command, 7-40, 9-44
- Unit Control Block
 - See UCB
- UNLOCK command, 4-78 to 4-79, 9-45
- User
 - displaying information, 8-8
 - types, 4-9, 4-83
- User File Directory
 - See UFD
- User Identification Code
 - See UIC
- Utility
 - using from DCL, 1-18
- VAX/VMS
 - DCL, 1-19
- Verify Utility
 - See VFY utility
- Version number
 - file, 4-3
 - changing, 4-47
 - default, 4-4
- VFY utility, 4-87
- Volume, 4-1, 5-1, 5-3, 5-4
 - backup, 5-73
 - density
 - changing, 5-70
 - setting, 5-59
 - dismounting, 5-53
 - file
 - default number, 5-62
 - maximum number
 - changing, 5-71
 - setting, 5-62
- Volume (Cont.)
 - Files-11, 5-3
 - backing up, 5-73
 - formatting, 5-57
 - level 401, 5-62
 - level 402, 5-62
 - restoring, 5-73
 - foreign, 5-3, 5-8, 5-38
 - label, 5-37, 5-58
 - changing, 5-71
 - magnetic tape, 5-3
 - mounting, 5-36
 - owner
 - changing, 5-71
 - specifying, 5-42, 5-63
 - protection, 4-9
 - changing, 5-71
 - code, 4-10
 - setting, 5-40
 - write-protecting, 5-42
- Volume Home Block, 5-57
 - changing, 5-70
 - displaying information, 5-71
- Volume ID, 5-38, 5-48
- Volume Identifier
 - See Volume ID
- VT100, 3-1
- VT200-series, 3-1
 - terminal keypad, 3-15
- VT52, 3-1
- Wildcard
 - asterisk (*), 3-14, 4-6 to 4-8
 - file specification, 4-6
 - percent sign (%), 3-14, 4-8
- WK:, 5-6
- Write access, 4-10, 4-84

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

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 user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital

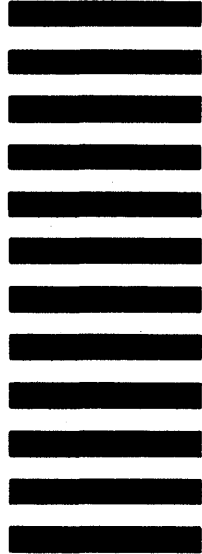


No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here