

**RSX-11M-PLUS**  
**Command Language Manual**

Order No. AA-FD04A-TC

RSX-11M-PLUS Version 3.0

First Printing, October 1979  
Revised, November 1981  
Updated, 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.

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	<b>digital</b>

ZK2642

---

#### 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 800-267-6215

##### 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	xi
SUMMARY OF TECHNICAL CHANGES	xv
CHAPTER 1	HOW TO USE DCL
1.1	RSX-11M-PLUS AND DCL: BASIC CONCEPTS . . . . . 1-1
1.2	DCL COMMAND DESCRIPTIONS . . . . . 1-2
1.2.1	The SET and SHOW Commands . . . . . 1-3
1.2.2	Terminal Operations . . . . . 1-3
1.2.3	Handling Files . . . . . 1-4
1.2.4	Handling Devices and Volumes . . . . . 1-5
1.2.5	LINK and LIBRARY . . . . . 1-6
1.2.6	Running Tasks . . . . . 1-6
1.2.7	System Control . . . . . 1-7
1.3	THE DCL COMMAND LINE . . . . . 1-7
1.3.1	Prompting . . . . . 1-8
1.3.2	Qualifiers . . . . . 1-9
1.3.3	HELP . . . . . 1-10
1.3.4	MCR and DCL Commands . . . . . 1-12
1.3.5	Abbreviations . . . . . 1-12
1.3.6	Numbers and Dates . . . . . 1-13
1.3.7	Multiple Parameters . . . . . 1-14
1.3.8	Underscore Character . . . . . 1-14
1.3.9	Colon and Equal Sign . . . . . 1-14
1.3.10	Quoting Strings . . . . . 1-15
1.3.11	ANSI File Names . . . . . 1-15
1.3.12	DECnet and DCL . . . . . 1-16
1.3.12.1	SET HOST . . . . . 1-16
1.3.12.2	SHOW HOST . . . . . 1-17
1.3.13	Command Line Continuation . . . . . 1-18
1.3.14	Comments in Command Lines . . . . . 1-18
1.3.15	SET DEBUG . . . . . 1-19
1.3.16	Errors . . . . . 1-20
1.4	USING THE QUEUE MANAGER: PRINT AND BATCH JOBS . . . . . 1-21
1.5	INDIRECT COMMAND FILES IN DCL . . . . . 1-22
1.6	DCL AND RSX-11M-PLUS UTILITIES AND SYSTEM TASKS . . . . . 1-22
1.7	RSX-11M-PLUS DCL AND VAX/VMS DCL . . . . . 1-23
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 . . . . . 2-4
2.5	SHOW QUEUE . . . . . 2-4
2.5.1	SHOW PROCESSOR . . . . . 2-4
2.6	SHOW ASSIGNMENTS AND SHOW LOGICALS . . . . . 2-5
2.7	SET AND SHOW DEFAULTS . . . . . 2-5
2.8	SHOW TASKS . . . . . 2-6
2.9	SHOW USERS . . . . . 2-7
2.10	SET PROTECTION . . . . . 2-7
2.11	SET AND SHOW SYSTEM/DIRECTORY AND LIBRARY/DIRECTORY . . . . . 2-8
2.12	SET DEBUG . . . . . 2-9

# CONTENTS

CHAPTER 3	TERMINAL OPERATIONS	
3.1	LOGGING IN AND LOGGING OUT . . . . .	3-2
3.1.1	LOGIN . . . . .	3-4
3.1.2	LOGOUT . . . . .	3-9
3.1.3	ACNT, the Account File Maintenance Program . . . . .	3-10
3.1.3.1	SET PASSWORD . . . . .	3-11
3.1.3.2	Resource Accounting on RSX-11M-PLUS . . . . .	3-12
3.2	THE KEYBOARD . . . . .	3-13
3.2.1	RETURN and Command Line Length . . . . .	3-13
3.2.2	Line Terminators . . . . .	3-14
3.2.3	DELETE Versus BACK SPACE . . . . .	3-14
3.2.4	CAPS LOCK and SHIFT . . . . .	3-15
3.2.5	REPEAT . . . . .	3-15
3.2.6	NO SCROLL or HOLD SCREEN . . . . .	3-16
3.2.7	Other Keys . . . . .	3-16
3.3	KEYPADS . . . . .	3-18
3.4	THE CTRL KEY AND CONTROL CHARACTERS . . . . .	3-20
3.4.1	CTRL/C . . . . .	3-20
3.4.2	CTRL/S and CTRL/Q . . . . .	3-21
3.4.3	CTRL/O . . . . .	3-21
3.4.4	CTRL/Z . . . . .	3-22
3.4.5	CTRL/U . . . . .	3-22
3.4.6	CTRL/R . . . . .	3-23
3.4.7	CTRL/X . . . . .	3-23
3.4.8	CTRL/L . . . . .	3-23
3.4.9	Other Control Characters . . . . .	3-24
3.5	HELP . . . . .	3-25
3.5.1	Help File Format: Writing a Help File . . . . .	3-28
3.6	BROADCAST . . . . .	3-32
3.7	REQUEST . . . . .	3-36
3.8	SET AND SHOW TERMINAL . . . . .	3-37
3.8.1	SET TERMINAL . . . . .	3-38
3.8.2	SHOW TERMINAL . . . . .	3-53

CHAPTER 4	HANDLING FILES	
4.1	FILE OWNERSHIP AND LOCATION . . . . .	4-2
4.2	FILE SPECIFICATION . . . . .	4-2
4.3	DEFAULTS IN FILE SPECIFICATIONS . . . . .	4-4
4.4	WILDCARDS IN FILE SPECIFICATIONS . . . . .	4-6
4.5	DATE-RELATED QUALIFIERS . . . . .	4-9
4.6	PROTECTION . . . . .	4-9
4.6.1	File Protection . . . . .	4-10
4.6.2	Directory Protection . . . . .	4-13
4.6.3	Volume Protection . . . . .	4-13
4.7	CREATING DIRECTORIES AND SEQUENTIAL FILES . . . . .	4-15
4.7.1	CREATE . . . . .	4-15
4.7.2	CREATE/DIRECTORY . . . . .	4-17
4.7.3	DELETE/DIRECTORY . . . . .	4-20
4.7.4	EDIT . . . . .	4-21
4.7.4.1	EDIT[/EDT] . . . . .	4-22
4.7.4.2	EDIT/SLP . . . . .	4-26
4.7.4.3	EDIT/EDI and EDIT/editor . . . . .	4-28
4.8	MAINTAINING A DIRECTORY . . . . .	4-30
4.8.1	DIRECTORY . . . . .	4-31
4.8.2	DELETE . . . . .	4-40
4.8.3	PURGE . . . . .	4-44
4.8.4	COPY . . . . .	4-47
4.8.5	RENAME . . . . .	4-52
4.8.6	CONVERT . . . . .	4-55
4.9	OTHER FILE-RELATED COMMANDS . . . . .	4-61
4.9.1	TYPE . . . . .	4-62
4.9.2	APPEND . . . . .	4-65
4.9.3	DIFFERENCES . . . . .	4-68

# CONTENTS

4.9.4	UNLOCK . . . . .	4-74
4.10	PRINT AND QUEUE MANAGER COMMANDS . . . . .	4-75
4.10.1	PRINT . . . . .	4-76
4.10.2	Queue Manager Commands for the User . . . . .	4-85
4.10.2.1	DELETE Command . . . . .	4-86
4.10.2.2	SHOW QUEUE . . . . .	4-88
4.10.2.3	SHOW PROCESSOR . . . . .	4-91
4.10.2.4	SET QUEUE . . . . .	4-92
4.10.2.5	STOP/ABORT Command . . . . .	4-96
4.10.2.6	HOLD . . . . .	4-97
4.10.2.7	RELEASE . . . . .	4-98
4.11	SET AND SHOW . . . . .	4-98
4.11.1	SET DEFAULT . . . . .	4-100
4.11.2	SHOW DEFAULT . . . . .	4-104
4.11.3	SET FILE . . . . .	4-105
4.11.4	SET UIC . . . . .	4-108
4.11.5	SHOW UIC . . . . .	4-110
4.11.6	SET PROTECTION . . . . .	4-111
4.11.7	SET PROTECTION/[NO]DEFAULT . . . . .	4-115
4.11.8	SHOW PROTECTION . . . . .	4-118

## CHAPTER 5 HANDLING DEVICES AND VOLUMES

5.1	DEVICES ON RSX-11M-PLUS SYSTEMS . . . . .	5-1
5.1.1	Volumes and Devices . . . . .	5-5
5.1.2	Physical Devices, Pseudo Devices, LUNs, and Logical Names . . . . .	5-6
5.1.2.1	Pseudo Devices . . . . .	5-6
5.1.2.2	LUNs . . . . .	5-8
5.1.2.3	Logical Device Names . . . . .	5-9
5.1.3	Public, Shareable, Private, and Unowned Devices and Mounted Volumes . . . . .	5-10
5.1.4	Preparing a Scratch Disk for Use . . . . .	5-11
5.1.5	Preparing a Scratch ANSI Magnetic Tape for Use . . . . .	5-13
5.1.6	Summary of Device and Volume Use Commands . . . . .	5-13
5.2	LOGICAL NAMES . . . . .	5-14
5.2.1	Logical Name Tables . . . . .	5-15
5.2.1.1	Displaying Logical Name Table Entries . . . . .	5-15
5.2.2	How To Create and Delete Logical Names . . . . .	5-16
5.2.3	Logical Name Translation . . . . .	5-17
5.2.3.1	Iterative Translation . . . . .	5-17
5.3	ANALYZE/MEDIA . . . . .	5-19
5.4	ASSIGN . . . . .	5-26
5.4.1	For RSX-11M-PLUS Systems Without Extended Logical Name Support . . . . .	5-26
5.4.2	For RSX-11M-PLUS Systems With Extended Logical Name Support . . . . .	5-29
5.4.3	ASSIGN/REDIRECT . . . . .	5-34
5.5	DEASSIGN . . . . .	5-37
5.6	DEFINE . . . . .	5-41
5.7	ALLOCATE . . . . .	5-45
5.8	DEALLOCATE . . . . .	5-48
5.9	MOUNT . . . . .	5-50
5.9.1	Mounting Magnetic Tapes and Tape Sets . . . . .	5-62
5.10	DISMOUNT . . . . .	5-68
5.11	INITIALIZE . . . . .	5-73
5.11.1	INITIALIZE/UPDATE . . . . .	5-86
5.12	BACKUP . . . . .	5-90
5.12.1	On-Line BACKUP Diskette, Disk, and Tape Device Information . . . . .	5-90
5.12.2	The Backup Command Line . . . . .	5-91
5.12.2.1	BACKUP Command Qualifiers . . . . .	5-91
5.12.2.2	Selective Backup and Restore Qualifiers . . . . .	5-93
5.12.2.3	Initialization Qualifiers . . . . .	5-95
5.12.2.4	Tape Control . . . . .	5-98

## CONTENTS

5.12.2.5	Verification Qualifiers . . . . .	5-99
5.12.2.6	Display Qualifiers . . . . .	5-99
5.12.2.7	Disk Processing Qualifiers . . . . .	5-100
5.12.3	Examples . . . . .	5-101
5.12.4	Messages . . . . .	5-103
5.12.4.1	BRU Messages . . . . .	5-105
5.13	SET AND SHOW . . . . .	5-128
5.13.1	SHOW ASSIGNMENTS and SHOW LOGICALS . . . . .	5-129
5.13.2	SET DEVICE . . . . .	5-132
5.13.3	SHOW DEVICES . . . . .	5-135

## CHAPTER 6 LINK AND LIBRARY COMMANDS

6.1	LINKING YOUR TASK WITH THE TASK BUILDER . . . . .	6-1
6.2	REVIEW OF THE PROGRAMMING PROCESS . . . . .	6-2
6.2.1	The Steps You Take When Programming . . . . .	6-2
6.2.2	The Source File . . . . .	6-3
6.2.3	The Object File . . . . .	6-3
6.2.4	The Task File . . . . .	6-3
6.3	INPUT FILES FOR THE TASK BUILDER . . . . .	6-3
6.3.1	Object Files as Input to the Task Builder . . . . .	6-4
6.3.2	Library Files as Input to the Task Builder . . . . .	6-4
6.3.3	Overlay Description Files as Input to the Task Builder . . . . .	6-5
6.3.4	Command Files as Input to the Task Builder . . . . .	6-6
6.4	THE TASK BUILDER FUNCTIONS . . . . .	6-6
6.4.1	Creating a Task . . . . .	6-7
6.4.2	Arranging Addresses . . . . .	6-7
6.4.3	Finding Symbols and Their Definitions . . . . .	6-8
6.5	THE LINK COMMAND . . . . .	6-10
6.5.1	LINK . . . . .	6-15
6.6	LIBRARY . . . . .	6-38
6.6.1	LIBRARY/COMPRESS . . . . .	6-40
6.6.2	LIBRARY/CREATE . . . . .	6-42
6.6.3	LIBRARY/DELETE . . . . .	6-45
6.6.4	LIBRARY/EXTRACT . . . . .	6-46
6.6.5	LIBRARY/INSERT . . . . .	6-47
6.6.6	LIBRARY/LIST . . . . .	6-50
6.6.7	LIBRARY/REMOVE . . . . .	6-54
6.6.8	LIBRARY/REPLACE . . . . .	6-55
6.6.9	Error Messages from LIBRARY . . . . .	6-57
6.7	SET AND SHOW . . . . .	6-63
6.7.1	SET GROUPFLAGS . . . . .	6-63
6.7.2	SHOW GROUPFLAGS . . . . .	6-65

## 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 . . . . .	7-8
7.2.2	Running and Scheduling Installed Tasks . . . . .	7-12
7.3	ABORT . . . . .	7-17
7.4	CONTINUE . . . . .	7-21
7.5	CANCEL . . . . .	7-23
7.6	STOP/BLOCK . . . . .	7-24
7.7	START . . . . .	7-26
7.7.1	START/UNBLOCK . . . . .	7-27
7.8	INSTALL . . . . .	7-28
7.9	REMOVE . . . . .	7-39
7.10	FIX . . . . .	7-41
7.11	UNFIX . . . . .	7-43
7.12	ASSIGN/TASK . . . . .	7-44
7.13	DEBUG . . . . .	7-46

## CONTENTS

7.14	SET AND SHOW . . . . .	7-47
7.14.1	SET PARTITION . . . . .	7-48
7.14.2	SHOW PARTITIONS . . . . .	7-52
7.14.3	SHOW TASKS . . . . .	7-55
7.14.3.1	SHOW TASKS/ACTIVE . . . . .	7-55
7.14.3.2	SHOW TASKS/INSTALLED . . . . .	7-60
7.14.3.3	SHOW TASKS/INSTALLED/DEVICE . . . . .	7-63
7.14.3.4	SHOW TASKS/DYNAMIC . . . . .	7-64
7.14.3.5	SHOW TASKS:taskname/DYNAMIC . . . . .	7-64
7.14.3.6	SHOW TASKS/ACTIVE/DYNAMIC . . . . .	7-66
7.14.3.7	SHOW TASKS/LOGICAL_UNITS . . . . .	7-68
7.14.4	SET PRIORITY . . . . .	7-71
7.14.5	SHOW CLOCK_QUEUE . . . . .	7-73
7.14.6	SHOW COMMON . . . . .	7-74

### CHAPTER 8          SYSTEM CONTROL

8.1	SET AND SHOW . . . . .	8-1
8.1.1	SET DAYTIME . . . . .	8-2
8.1.2	SHOW [DAY]TIME . . . . .	8-4
8.1.3	SET SYSTEM . . . . .	8-5
8.1.4	SHOW SYSTEM . . . . .	8-8
8.2	SYSTEM AND LIBRARY DIRECTORIES . . . . .	8-10
8.2.1	SET LIBRARY/DIRECTORY . . . . .	8-11
8.2.2	SHOW LIBRARY . . . . .	8-12
8.3	SHOW USERS . . . . .	8-13
8.4	SHOW MEMORY . . . . .	8-15

### CHAPTER 9          QUICK REFERENCE

ABORT . . . . .	9-1
ALLOCATE . . . . .	9-2
ANALYZE/CRASH_DUMP . . . . .	9-2
ANALYZE/MEDIA . . . . .	9-3
APPEND . . . . .	9-4
ASSIGN . . . . .	9-4
ASSIGN/QUEUE . . . . .	9-4
ASSIGN/REDIRECT . . . . .	9-5
ASSIGN/TASK . . . . .	9-5
BACKUP . . . . .	9-5
BROADCAST . . . . .	9-7
CANCEL . . . . .	9-7
CONTINUE . . . . .	9-7
CONVERT . . . . .	9-7
COPY . . . . .	9-8
CREATE . . . . .	9-9
CREATE/DIRECTORY . . . . .	9-9
DEALLOCATE . . . . .	9-9
DEASSIGN . . . . .	9-9
DEASSIGN/QUEUE . . . . .	9-10
DEBUG . . . . .	9-10
DEFINE . . . . .	9-10
DELETE . . . . .	9-11
DELETE/DIRECTORY . . . . .	9-11
DELETE/ENTRY . . . . .	9-11
DELETE/JOB . . . . .	9-12
DELETE/processortype . . . . .	9-12
DELETE/QUEUE . . . . .	9-12
DIFFERENCES . . . . .	9-12
DIRECTORY . . . . .	9-13
DISMOUNT . . . . .	9-14
EDIT[/EDT] . . . . .	9-14
EDIT/SLP . . . . .	9-14
EDIT/EDI and EDT/editor . . . . .	9-15

# CONTENTS

FIX . . . . .	9-15
HELP . . . . .	9-16
HOLD/ENTRY . . . . .	9-16
HOLD/JOB . . . . .	9-16
INITIALIZE . . . . .	9-16
INITIALIZE/processor type (Input) . . . . .	9-17
INITIALIZE/processor type (Output) . . . . .	9-18
INITIALIZE/QUEUE . . . . .	9-18
INITIALIZE/UPDATE . . . . .	9-18
INSTALL . . . . .	9-19
LIBRARY . . . . .	9-19
LIBRARY/COMPRESS . . . . .	9-20
LIBRARY/CREATE . . . . .	9-20
LIBRARY/DELETE . . . . .	9-20
LIBRARY/EXTRACT . . . . .	9-20
LIBRARY/INSERT . . . . .	9-21
LIBRARY/LIST . . . . .	9-21
LIBRARY/REMOVE . . . . .	9-21
LIBRARY/REPLACE . . . . .	9-21
LINK . . . . .	9-22
LOGIN . . . . .	9-23
LOGOUT . . . . .	9-23
MCR . . . . .	9-24
MOUNT . . . . .	9-24
PRINT . . . . .	9-25
PURGE . . . . .	9-26
RELEASE/ENTRY . . . . .	9-26
RELEASE/JOB . . . . .	9-27
REMOVE . . . . .	9-27
RENAME . . . . .	9-27
REQUEST . . . . .	9-27
RUN Installed Task . . . . .	9-28
RUN Uninstalled Task . . . . .	9-28
SET [DAY]TIME . . . . .	9-29
SET DEBUG . . . . .	9-29
SET DEFAULT . . . . .	9-29
SET DEVICE . . . . .	9-29
SET FILE . . . . .	9-30
SET GROUPFLAGS . . . . .	9-30
SET HOST . . . . .	9-30
SET LIBRARY/DIRECTORY . . . . .	9-31
SET [NO]PARTITION . . . . .	9-31
SET PASSWORD . . . . .	9-31
SET PRIORITY . . . . .	9-31
SET PROTECTION . . . . .	9-31
SET PROTECTION/[NO]DEFAULT . . . . .	9-32
SET QUEUE/ENTRY . . . . .	9-32
SET QUEUE/JOB . . . . .	9-32
SET SYSTEM . . . . .	9-33
SET TERMINAL . . . . .	9-33
SET UIC . . . . .	9-35
SHOW ACCOUNTING . . . . .	9-35
SHOW ASSIGNMENTS . . . . .	9-36
SHOW CACHE . . . . .	9-36
SHOW CLOCK_QUEUE . . . . .	9-36
SHOW COMMON . . . . .	9-36
SHOW [DAY]TIME . . . . .	9-37
SHOW DEFAULT . . . . .	9-37
SHOW DEVICES . . . . .	9-37
SHOW HOST . . . . .	9-37
SHOW LIBRARY . . . . .	9-37
SHOW LOGICALS . . . . .	9-38
SHOW MEMORY . . . . .	9-38
SHOW PARTITIONS . . . . .	9-38
SHOW PROCESSOR . . . . .	9-38
SHOW PROTECTION . . . . .	9-39

## CONTENTS

SHOW QUEUE . . . . .	9-39
SHOW SYSTEM . . . . .	9-40
SHOW TASKS . . . . .	9-40
SHOW TASKS/DYNAMIC . . . . .	9-40
SHOW TERMINAL . . . . .	9-41
SHOW UIC . . . . .	9-42
SHOW USERS . . . . .	9-43
START . . . . .	9-43
START/processor type . . . . .	9-43
START/QUEUE . . . . .	9-44
START/QUEUE/MANAGER . . . . .	9-44
START/UNBLOCK . . . . .	9-44
STOP/ABORT . . . . .	9-44
STOP/BLOCK . . . . .	9-44
STOP/processor type . . . . .	9-45
STOP/QUEUE . . . . .	9-45
STOP/QUEUE/MANAGER . . . . .	9-45
SUBMIT . . . . .	9-45
TYPE . . . . .	9-46
UNFIX . . . . .	9-46
UNLOCK . . . . .	9-47

### CHAPTER 10      COMMON ERROR MESSAGES

10.1	DCL ERROR MESSAGES . . . . .	10-1
10.2	TASK TERMINATION MESSAGES . . . . .	10-18

### APPENDIX A      FUNCTIONS INITIATED BY DCL COMMANDS

#### INDEX

#### FIGURES

3-1	VT200-Series Keyboard and Keypad . . . . .	3-1
4-1	File Organization on a Volume . . . . .	4-1
4-2	Job Flag Page . . . . .	4-78
4-3	File Flag Page . . . . .	4-79
6-1	Module Address Assignment . . . . .	6-8

#### TABLES

3-1	Keyboard and Keypad Characters . . . . .	3-16
3-2	Display of Terminal Attributes from SHOW TERMINAL . . . . .	3-55
4-1	File Types . . . . .	4-5
5-1	Devices on RSX-11M-PLUS Systems . . . . .	5-2
5-2	Pseudo Devices on RSX-11M-PLUS . . . . .	5-7
6-1	Task Builder Options . . . . .	6-31
7-1	Base and Size Specifications for SET PARTITION . . . . .	7-49
7-2	Task Status Flags . . . . .	7-58
A-1	Functions Initiated By DCL Commands . . . . .	A-1





## PREFACE

### MANUAL OBJECTIVES

The RSX-11M-PLUS Command Language Manual provides information needed to do work on a RSX-11M-PLUS system. Important system concepts are introduced and explained, including how to use the DIGITAL Command Language (DCL) to communicate with the system. DCL is based on English words and is designed for ease in issuing commands to the system. The manual also gives instruction in using the EDT editor and many other useful system facilities.

### INTENDED AUDIENCE

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

### ASSOCIATED MANUALS

If you are entirely new to computers, or to DCL, or to the RSX family of operating systems, you should read the Introduction to RSX-11M-PLUS and follow its instructions before proceeding to the RSX-11M-PLUS Command Language Manual.

Other manuals related to RSX-11M-PLUS are described in the RSX-11M-PLUS Information Directory and Master Index.

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.

## PREFACE

### STRUCTURE OF THIS MANUAL

The RSX-11M-PLUS Command Language 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.

- Chapter 1    How to Use DCL, introduces the basic concepts of DCL and includes many hints on using DCL.
- Chapter 2    The SET and SHOW Commands, lists the most common forms of SET and SHOW commands.
- Chapter 3    Terminal Operations, gives full coverage of terminal operations, including logging in and logging out, keyboard terminology and functions, setting and displaying terminal attributes, broadcasting messages, using the system HELP commands, and writing a help file.
- Chapter 4    Handling Files, goes into more detail about commands that list directories, create, delete, rename, copy, type, and print files. Wildcards and defaults are explained.
- Chapter 5    Handling Devices and Volumes, discusses peripheral devices and explains the system I/O terminology. The chapter explains the relationship 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 also described.
- Chapter 6    LINK and LIBRARY Commands, is for programmers. It documents the program development facilities included in RSX-11M-PLUS. The chapter explains the RSX Task Builder and the LIBRARY command, which is used to maintain libraries.
- Chapter 7    Running Tasks, 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 that abort tasks, place tasks in the clock queue, and display information about tasks in the system.
- Chapter 8    System Control, explains system-control commands including those which set the time and date, set and show certain system characteristics, and show memory utilization.
- Chapter 9    Quick Reference, is the alphabetical listing of all DCL commands and their formats, including comments that 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 elsewhere.
- Chapter 10   Common Error Messages, explains error messages common to several DCL commands. Other command-specific error messages are explained in the full command descriptions.
- Appendix A   Functions Initiated by DCL Commands, lists the relationship between DCL and the rest of the operating system.

## PREFACE

### CONVENTIONS USED IN THIS MANUAL

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

Convention	Meaning
.	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 a User Identification Code (UIC). The g is a group number and the m is a member number. Where a UIC is required, only one set of brackets is shown, as [g,m]. Where the UIC 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 function.
[directory]	This signifies a directory, which is the location of files. Most directories have 1- to 9-character names, but some are in the same [g,m] form as the UIC. Where a directory name is required, only one set of brackets is shown, as [directory]. Where the directory is optional, two sets of brackets are shown, as [[directory]].
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 also 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 a 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:QIX). The equal sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.

## PREFACE

### Convention

### Meaning

`filespec` A full file specification includes device, directory, file name, file type, and version number, as in the following example:

```
DU1:[COTTONTAL]HIPPIITY.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.

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.

**RET**

A symbol with a 2- to 6-character abbreviation indicates that you are to press the corresponding key on your terminal. For example, **RET** indicates that you are to press the **RETURN** key, and **DEL** means that you are to press the **DELETE** key.

**{A I B}**

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

**CTRL/a**

The symbol **CTRL/a** means that you are to press the key marked **CTRL** while pressing another key. Thus, **CTRL/Z** indicates that you are to press the **CTRL** key and the **Z** key together in this fashion. **CTRL/Z** is echoed on your terminal as ^Z. Not all control characters echo. See Chapter 3 for more information.

See Chapter 1, for more information on DCL conventions.

## SUMMARY OF TECHNICAL CHANGES

### NEW COMMANDS AND QUALIFIERS

ANALYZE/MEDIA[/qualifier[s]]  
    /ALLOCATE=label  
    /BADBLOCKS  
    /BADBLOCKS/NOEXERCISE  
    /[NO]EXERCISE[=(n,m)]  
    /OVERRIDE  
    /RETRY  
    /SHOW

New command. 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.

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

New qualifiers.

ASSIGN[/qualifier[s]]  
    /GROUP:[g]  
    /FINAL

New qualifiers.

COPY[/qualifier[s]]  
    /ALLOCATION:n[.]  
    /NOWARNINGS  
    /OVERLAY  
    /PRESERVE\_DATE  
    /REWIND  
    /SHARED

New qualifiers.

CREATE/DIRECTORY[/qualifier[s]]  
    /NOWARNINGS  
    /OWNER\_UIC:[uic]

New qualifiers.

## SUMMARY OF TECHNICAL CHANGES

DEASSIGN[/qualifier[s]]  
/GROUP[:g]

New qualifier.

DEFINE[/qualifier[s]]  
/GLOBAL  
/GROUP[:g]  
/LOCAL  
/LOGIN  
/SYSTEM  
/TERMINAL:ttnn:  
/FINAL

New command. Equates a logical name to a physical device name, to all or part of a file specification, or to another logical name.

DELETE[/qualifier[s]]  
/NOWARNINGS

New qualifier.

DELETE/DIRECTORY

New command. Deletes a directory on a Files-11 volume and removes its name from the volume's Master File Directory (MFD).

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

New qualifiers.

DISMOUNT[/qualifier[s]]  
/TERMINAL:ttnn:

New qualifier.

INITIALIZE[/qualifier[s]]  
/BAD\_BLOCKS:arg  
NOAUTOMATIC  
(AUTOMATIC,MANUAL)  
(OVERRIDE,MANUAL)  
/DENSITY:arg  
HIGH  
LOW  
/LABEL:VOLUME\_ACCESSIBILITY:"c"  
/PROFESSIONAL

New qualifiers.

INITIALIZE/UPDATE[/qualifier[s]]  
/PROFESSIONAL  
/LABEL:newvolumelabel

New qualifiers.

INSTALL[/qualifier[s]]  
/[NO]INTERPRETER  
/[NO]RESIDENT HEADER  
/[NO]TRANSLATION\_ROUTINE:n  
/[NO]WRITE\_BACK

New qualifiers.

## SUMMARY OF TECHNICAL CHANGES

LINK[/qualifier[s]]  
/CODE:(arg[s])  
CLI  
FAST\_MAP  
/[NO]PRINT

New qualifiers.

MOUNT[/qualifier[s]]  
/PROCESSOR:arg  
UNIQUE  
/OVERRIDE:arg  
IDENTIFICATION

New qualifiers.

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

New qualifier.

PURGE[/qualifier[s]]  
/NOWARNINGS

New qualifier.

RENAME[/qualifier[s]]  
/NOWARNINGS

New qualifier.

REMOVE[/qualifier]  
/TRANSLATION\_ROUTINE

New qualifier.

SET DEFAULT[/qualifier]  
/[NO]NAMED\_DIRECTORY]

New qualifier.

SET FILE[/qualifier[s]]  
/ENTER:synonym\_filespec  
/NOWARNINGS  
/REMOVE  
/REWIND  
/TRUNCATE  
/END\_OF\_FILE:(BLOCK:n,BYTE:n)

New command. Establishes certain file characteristics.

SET HOST

New command. Connects your terminal to another system.

SET PASSWORD

New command. Allows users to change their passwords.

SET PROTECTION[/qualifier]  
/[NO]DEFAULT

New qualifier.

## SUMMARY OF TECHNICAL CHANGES

```
SET SYSTEM[/qualifier[s]]
    /NETWORK_UIC
    /POOL/LIMITS:arg
        HIGH=n
        LOW=n
        MINIMUM_SIZE=n
        TASK_PRIORITY=n
```

New qualifiers.

```
SET TERMINAL[/qualifier[s]]
    /[NO]CONTROL=C
    /INQUIRE
    /DTC01
    /LA50
    /LA210
    /LN03
    /LQP02
    /LQP03
    /PRO_SERIES
    /VT200_SERIES
    /[NO]ADVANCED_VIDEO
    /[NO]ANSI_CRT
    /[NO]BLOCK_MODE
    /[NO]EDIT_MODE
    /[NO]HOSTSYNC
    /PRINTER_PORT
    /[NO]REGIS
    /[NO]SOFT_CHARACTERS
    /[NO]TRANSLATION_ROUTINE:n
    /[NO]TTSYNC
    /[NO]PARITY:arg
        ODD
        EVEN
```

New qualifiers.

SET UIC

New command. Changes your User Identification Code (UIC).

```
SHOW ASSIGNMENTS[/qualifier[s]]
    /GROUP[:g]
    /TERMINAL:ttnn:
```

New qualifiers.

SHOW HOST

Displays the name of the processor to which your terminal is currently connected.

```
SHOW LOGICALS[/qualifier[s]]
    /ALL
    /GLOBAL
    /GROUP[:g]
    /LOCAL
    /LOGIN
    /SYSTEM
    /TERMINAL:ttnn
```

New command. Displays at your terminal all local and login logical name assignments.



## SUMMARY OF TECHNICAL CHANGES

### SHOW PROTECTION

New command. Displays your personal default file protection code.

SHOW SYSTEM[/qualifier[s]]  
          /NETWORK\_UIC  
          /POOL/LIMITS

New qualifiers.

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

New qualifier.

SHOW TERMINAL[/qualifier[s]]  
          /[NO]CONTROL=C  
          /[NO]DTC01  
          /[NO]LA50  
          /[NO]LA210  
          /[NO]LN03  
          /[NO]LQP02  
          /[NO]LQP03  
          /[NO]PRO\_SERIES  
          /[NO]VT200\_SERIES  
          /[NO]ADVANCED\_VIDEO  
          /[NO]ANSI\_CRT  
          /[NO]BLOCK\_MODE  
          /[NO]EDIT\_MODE  
          /[NO]HOSTSYNC  
          /PRINTER\_PORT  
          /[NO]REGIS  
          /[NO]SOFT\_CHARACTERS  
          /[NO]TTSYNC

New qualifiers.

### SHOW UIC

New command. Displays your User Identification Code (UIC).

## SUMMARY OF TECHNICAL CHANGES

### NEW DEVICE SUPPORT

The following table lists the new devices supported by RSX-11M-PLUS.

#### New terminal types:

DTC01	LA50
LA210	LN03
LQP02	LQP03
VT200_Series	PRO_Series

#### New disk device:

RD52

#### New magnetic tape devices:

TK25	TK50
TU80	

## CHAPTER 1

### HOW TO USE DCL

DCL is the DIGITAL Command Language. DCL gives users an extensive set of commands for dealing with the operating system. DCL commands provide 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-PLUS. This book 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, the abbreviation SET TERM/LOW will also set a terminal to lowercase, but SET TERM/LO will not work because there is also a command SET TERMINAL/LOCAL. See Section 1.3 for more information on how DCL commands are parsed.

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

#### 1.1 RSX-11M-PLUS AND DCL: BASIC CONCEPTS

RSX-11M-PLUS 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 that invoke most system tasks and utilities and 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-PLUS operating systems. MCR also provides commands that set and display certain system characteristics. MCR provides the most direct interface with the RSX-11M-PLUS operating system.

## HOW TO USE DCL

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

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

DCL is the command language used on many DIGITAL operating systems. In particular, RSX-11M-PLUS DCL is designed for compatibility with future releases of Micro/Rsx, RSX-11M, and VAX/VMS DCL. DCL on RSX-11M-PLUS systems is a CLI task that translates DCL commands into MCR commands for execution by the system. The DCL command SET DEBUG displays the MCR translation for any DCL command on your terminal. See Section 1.3.15 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 utilities functions are available from DCL.

Because RSX-11M-PLUS 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 user if some feature described in this manual does not appear to be available at your installation.

### 1.2 DCL COMMAND DESCRIPTIONS

This manual includes seven functional groupings of commands, each grouping constituting a chapter. Each functional grouping contains an introduction to the important concepts of the functional group and fully detailed descriptions of each command.

Command descriptions can have five parts:

1. A brief statement of the function of the command.
2. A 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 Chapter 10.

## HOW TO USE DCL

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

Nonprivileged commands are those commands needed by all users in everyday use of the system. 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.

The following sections (1.2.1 through 1.2.7) briefly describe how the seven functional groupings of commands are organized, how the commands work, and in which chapters the commands appear.

### 1.2.1 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. The SET and SHOW commands are the last commands described in each chapter. For example, the SET DEVICES and SHOW DEVICES commands are described in full at the end of Chapter 5; and the SET TERMINAL and SHOW TERMINAL commands are described at the end of Chapter 3.

The SET and SHOW command and its function are considered to each 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.

### 1.2.2 Terminal Operations

All commands directly related to terminal operations are described in Chapter 3.

These commands enable you to do the following operations:

Operation	DCL Command
Log you in to the computer	LOGIN
Log you out of the computer	LOGOUT
Invoke on-line help	HELP
Send messages to other terminals	BROADCAST
Send messages to the console terminal	REQUEST
Set various characteristics for your terminal	SET TERMINAL
Display your terminal's characteristics	SHOW TERMINAL

## HOW TO USE DCL

In addition, Chapter 3 includes a detailed description of the use of the terminal keyboard on RSX-11M-PLUS systems. All control characters (CTRL) and other important keys are explained.

### 1.2.3 Handling Files

Most information in RSX-11M-PLUS systems is in files in directories. These files and directories are located within Files-11 volumes.

All commands relating to files are described in Chapter 4.

These commands enable you to do the following operations:

Operation	DCL Command
Create a file	CREATE or EDIT
Create a directory	CREATE/DIRECTORY
Delete a directory	DELETE/DIRECTORY
Display information on files in a directory	DIRECTORY
Modify a file	EDIT
Delete a specified file	DELETE
Delete all but the latest version of a file	PURGE
Copy a file	COPY
Change the name of a file	RENAME
Move a file's records to another file	CONVERT
Print a file on your terminal	TYPE
Append a file to another file	APPEND
Compare two files	DIFFERENCES
Free locked files	UNLOCK
Queue a file for printing on a line printer	PRINT
Delete a job in the queue	DELETE/ENTRY DELETE/JOB
Display information about print and batch jobs	SHOW QUEUE
Display information about print and batch processors	SHOW PROCESSOR
Modify attributes of a print or batch job	SET QUEUE

## HOW TO USE DCL

Operation	DCL Command
Stop a current print or batch job from completing	STOP/ABORT
Block a print or batch job from processing	HOLD
Unblock a blocked print or batch job	RELEASE
Change your default directory	SET DEFAULT
Display your current default directory location	SHOW DEFAULT
Establish certain file attributes	SET FILE
Change your User Identification Code	SET UIC
Display your User Identification Code	SHOW UIC
Change the protection for a file	SET PROTECTION
Establish a default file protection code for all your files	SET PROTECTION/DEFAULT
Display your default file protection code	SHOW PROTECTION

### 1.2.4 Handling Devices and Volumes

Volumes are held on magnetic media, usually disks. The volume is in the RSX-11M-PLUS format, called Files-11. The disks are mounted on hardware devices. Devices can be public, available to all users; shareable, available to some users but not others; or private, available to a single user only.

All commands related to devices and volumes appear in Chapter 5.

These commands enable you to do the following operations:

Operation	DCL Command
Determine the number of bad blocks on a disk	ANALYZE/MEDIA
Assign a logical name to all or part of a file specification	ASSIGN or DEFINE
Delete logical name assignments	DEASSIGN
Redirect output from one physical device to another	ASSIGN/REDIRECT
Specify a device for private access	ALLOCATE
Free a private device	DEALLOCATE
Enable a volume for use on a system	MOUNT

## HOW TO USE DCL

Operation	DCL Command
Disable a mounted volume	DISMOUNT
Produce a volume in Files-11 format	INITIALIZE
Backup and restore a volume	BACKUP
Display logical assignments	SHOW ASSIGNMENTS or SHOW LOGICALS
Establish certain device characteristics	SET DEVICE
Display device information	SHOW DEVICES

### 1.2.5 LINK and LIBRARY

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

The SET GROUPFLAGS and SHOW GROUPFLAGS commands are also described.

### 1.2.6 Running Tasks

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

These commands enable you to do the following operations:

Operation	DCL Command
Initiate the execution of a task	RUN
End the execution of a specified command	ABORT
Resume execution of a suspended task	CONTINUE
Eliminate an entry in the clock queue	CANCEL
Enable a task for execution by the system	INSTALL
Block an installed task	STOP/BLOCK
Resume execution of a stopped task	START
Counteract the STOP/BLOCK command	START/UNBLOCK
Disable a task for execution by the system	REMOVE
Load and lock a task into memory	FIX
Unlock a fixed task in memory	UNFIX



## HOW TO USE DCL

Operation	DCL Command
Reassign an installed task's logical unit number to another device	ASSIGN/TASK
Debug a task	DEBUG
Create or eliminate partitions in memory	SET PARTITIONS
Display information about partitions in memory	SHOW PARTITIONS
Display information about active or installed tasks	SHOW TASKS
Alter an active task's priority	SET PRIORITY
Display information about tasks in the clock queue	SHOW CLOCK_QUEUE
Display information about resident commons installed in the system	SHOW COMMON

### 1.2.7 System Control

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

These commands enable you to do the following operations:

Operation	DCL Command
Set the date and time	SET DAYTIME
Display the date and time	SHOW DAYTIME
Establish certain system characteristics	SET SYSTEM
Display information about the system	SHOW SYSTEM
Establish the system library directory	SET LIBRARY/SYSTEM
Display the system library directory	SHOW LIBRARY
Display information about logged-in users	SHOW USERS
Display system resource information	SHOW MEMORY

### 1.3 THE DCL COMMAND LINE

A command consists of a command name (usually a verb) specifying 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 or prompt. Both qualifiers and parameters can take arguments.

Arguments are preceded by a colon (:), or an equals sign (=). This book documents only the colon.

Here is an example:

```
$ DIRECTORY/OUTPUT:UNCLE.DAT SOLO.* (RET)
```

In this case, the elements of the command line are as follows:

DIRECTORY is the command name.

- OUTPUT, which is preceded by a slash (/), is a qualifier. This qualifier indicates that you want the output from the DIRECTORY command to go somewhere other than your terminal screen, which is the default.
- UNCLE.DAT, which is preceded by a colon (:), is the argument to the /OUTPUT qualifier. This argument is the name of the file where you want the output from the command to go.
- SOLO.\*, which is preceded by a blank, is a parameter. This parameter is the name of the file(s) of which you are taking a directory listing. Most DCL parameters are file specifications, as is this one. The asterisk (\*) indicates a wildcard file type. See Section 4.4 for information on wildcards.

Note that after you type in a command, you must pass it to the operating system by pressing either the RETURN key or the ESC key. (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.) See Section 1.3.13 for a discussion of command lines that are 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 message rather than a prompt.)

### 1.3.1 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)
Old file name? BROWNS.STL (RET)
New file name? 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)
New file name?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.

### 1.3.2 Qualifiers

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

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 name, as in this example:

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

However, most command qualifiers can appear anywhere in the command line. They are also called "floating qualifiers." The following examples illustrate how command qualifiers can float:

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

or

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

or

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

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

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

or

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

or

```
$TYPE/TODAY (RET)
File(s)?/ (RET)
Qualifier?EXCLUDE:ILLYA.DAT;* (RET)
File(s)?*.DAT (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 a few cases, however, 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

## HOW TO USE DCL

qualifier cannot float. Other qualifiers to such commands can still float, however, as in this example:

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

which is the same as the following example:

```
$ CREATE/DIRECTORY/ALLOCATION:5 [JENA] (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 command

```
$ DELETE/LOG *.TXT;* (RET)
```

deletes all files with the type .TXT and prints a list of the deletions on your terminal, while the command

```
$ DELETE/NOLOG *.TXT;* (RET)
```

or

```
$ DELETE/-LOG *.TXT;* (RET)
```

simply deletes all the files without printing a list. (This action is in fact the default behavior, but the example is given to illustrate the use of the negative form of qualifiers.)

### 1.3.3 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.

For instance, if you need 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
```

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

To obtain help on the above qualifiers, type the following:

```
$ HELP TYPE qualifier
```

## HOW TO USE DCL

The HELP text consists of the command syntax, showing that TYPE accepts one or more file specifications and one or more qualifiers, followed by a brief explanation.

You can also type a HELP command naming the qualifier to get more information on that qualifier, as in this example:

```
$ HELPTYPE TODAY (RET)
```

```
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
```

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

To obtain help on the above qualifiers, type the following:

```
HELP TYPE qualifier
```

```
File(s)?
```

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 (?) followed by the subtopic. For example, after getting help on SET, you can also get help on a specific function of SET:

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

The SET command establishes or changes the following:

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

For information on the above commands, type HELP SET commandname.

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

```
Function? ? DEFAULT (RET)
```

```
SET DEFAULT [ddnn:][directory]
```

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

Function?

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

If you should decide after reading the help text that you have chosen the wrong command, enter a **CTRL/Z** in response to the prompt to end the execution of the command. (A **CTRL/Z** pressed before entering the command always cancels the 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 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.

### 1.3.4 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
```

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.

### 1.3.5 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.

## HOW TO USE DCL

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 you can abbreviate various commands. 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-PLUS version of DCL permits you to type the following:

```
$ SET VT100 (RET)
```

to achieve the same result. This 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 (or batch job). You should also see Section 1.7 for a discussion of compatibility between RSX-11M-PLUS DCL and other versions of DCL used on other operating systems.

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

Command	Brief Form	Command	Brief Form
ABORT	A	HELP	H
BROADCAST	B	HELP	?
COPY	C	LOGOUT	LO
DIRECTORY	D	LINK	L
REALLOCATE	DEAL	MACRO	M
REASSIGN	DEAS	PRINT	P
EDIT	E	RUN	R
FORTTRAN	F	SHOW	S
		TYPE	T

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

### NOTE

As new commands are added in future releases, abbreviations may change.

### 1.3.6 Numbers and Dates

DCL recognizes both octal and decimal numbers. You usually do not have to identify a number as octal or decimal, as DCL takes care of it. In rare cases, the command description directs you to add a decimal point to identify a decimal number.

DCL recognizes dates in two forms:

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

or

mm/dd/yy as in 06/21/85

System displays are always in the first format.

## 1.3.7 Multiple Parameters

Some commands permit you to enter a list of parameters instead of just one. If you are entering a list of parameters, each parameter must be set off by commas. For example,

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

causes a single file to be printed, while

```
$ PRINT JANE.TXT, CHRIS.TXT, MULP.TXT (RET)
```

causes three files to be printed. You have the option of including spaces 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 JANE.TXT, CHRIS.TXT, (RET)
File(s)? MULP.TXT (RET)
```

Some commands for program development accept a list of arguments to a single qualifier or parameter. In such cases, the list of arguments must be enclosed in parentheses, with the elements set off by commas. For example:

```
$ LINK/CODE:(PIC,FPP) HIYA (RET)
```

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

```
$ LINK/CODE:PIC HIYA (RET)
```

## 1.3.8 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. However, PRINT/FLAG PAGE will not work. You cannot include a space in a command field.

## 1.3.9 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 usually replace such colons with an equal sign (=), as in this example:

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

Colons in device names, such as DU1:, cannot be replaced by equal signs.

### 1.3.10 Quoting Strings

If you wish to include an exact string in a DCL command, put the string in quotation marks. For instance, the message

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

is broadcast as

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

while

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

is broadcast as

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

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

### 1.3.11 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, the file name "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?MM1:"CASH&." (RET)
```

If you are copying a file from a tape to a Files-11 disk, 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 (.).

### 1.3.12 DECnet and DCL

Your system may include the optional DECnet networking software. If so, you'll need the following information.

DECnet is a DIGITAL product that enables two or more systems to "talk" to each other. These systems are linked together physically to form a network. The purpose of a network is to allow the users on different systems to share information and resources. See the Introduction to DECnet for more information about this product.

Each system in a network is called a node. The system that you originally log in to is your local node. All other systems in a network are called remote nodes.

Some DCL commands accept DECnet node names as part of a file specification. They are the following:

APPEND	COPY	CREATE	DELETE
DIRECTORY	RENAME	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, SUBMIT, 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 name 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 [PETER].

```
$ TYPE PRINCE::DB2:[PETER]ROMAN.TXT (RET)
```

All the usual rules about file and volume protection are maintained, of course. See Section 4.5.

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

```
$TYPE NEMO::"DISK$USERDISK:[DALTON]DESPERADO READERS.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.

#### 1.3.12.1 SET HOST

After you have logged 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 to 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 to is called a local node; all other systems in a network are called remote nodes.

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 located on your local node.

You can only use SET HOST to connect to one remote node 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 on your local node, then use SET HOST to connect to JUNE.

See the RSX DECnet Guide to User Utilities for a full explanation of this command.

**Example**

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

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

```
$LOGIN (RET)
Account or Name:Erin (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 to the remote system, using the name and password of your account on that system.

**1.3.12.2 SHOW HOST**

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

## HOW TO USE DCL

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-PLUS V3.0 BL24
```

This example indicates the display from SHOW HOST. The name of your current processor is TOOTSI, which is running Version 3.0 of the RSX-11M-PLUS operating system.

### 1.3.13 Command Line Continuation

Sometimes a command will not fit all on one line. In that case you can continue the line by using a hyphen (-). When you end a command line with a hyphen and a 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/FLAG_PAGE/AFTER:(04-JUN-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 RETURN does not enter the command. Only the second RETURN, with no hyphen, enters the command. The RETURN can be on a line by itself.

### 1.3.14 Comments in Command Lines

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

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

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

If the comment is within the command line, you need two exclamation points 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)
```

### 1.3.15 SET DEBUG

The SET DEBUG command displays the MCR translation of any DCL command.

#### Format

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

#### Command Qualifiers

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

#### Command Qualifiers

```
/[NO]EXECUTE
```

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

```
/FULL
```

The /FULL qualifier indicates that you want the SET DEBUG command 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.

**Example**

```

$ SET DEBUG (RET)
$ SHOW TIME (RET)
TIM
$ INITIALIZE/EXTENSION:200/PROTECTION:(SYS:RWED,OWN:RWED,GRO,WO) (RET)
Device? DR2 (RET)
Label? HOTROD (RET)
INI DR2:HOTROD/EXT=200./PRO=[RWED,RWED ,]
$ SET DEBUG/EXECUTE (RET)
$ SHO TIM (RET)
TIM
21:33:46 17-JUN-85
$ SET NODEBUG (RET)

```

In this example, the user first issued the SET DEBUG command. Then, the user typed SHOW TIME. DCL displayed TIM, which is the MCR command that displays the date and time, but no date and time is displayed because the /NOEXECUTE qualifier was in effect. Next, the user typed an INITIALIZE command and DCL displayed the MCR translation of the same command. Then, the user issued the SET DEBUG/EXECUTE command 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 issued the SET NODEBUG command to end the session with the SET DEBUG command.

**Notes**

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

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

You should not use the SET DEBUG command if your terminal is set to serial mode. Because DCL remains an active task while you are in this mode (which is also an active task) and your terminal can process only one task at a time, you may encounter unpredictable results when your terminal tries to process two tasks at the same time.

**1.3.16 Errors**

You can correct typing errors or change the line completely by using the DELETE key or 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.

## HOW TO USE DCL

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 the 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 is often all you need to do to eliminate 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 Chapter 10.

### 1.4 USING THE QUEUE MANAGER: PRINT AND BATCH JOBS

The RSX-11M-PLUS 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 Chapter 4. The description includes information on how to display and alter jobs in the print queues.

In addition, the Queue manager provides a batch processing facility. Batch jobs allow you to use the system without requiring you to be present. This means that jobs that take a long time to run, or otherwise tie up system facilities, can be run when there are fewer demands on the system, such as at night or on weekends.

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 and the

## HOW TO USE DCL

batch-processing subsystem. The manual includes sample print and batch jobs, full descriptions of commands to the batch processors, 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 batch processing, RSX-11M-PLUS provides the Indirect Command Processor (Indirect) as a 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 directives. You use these commands and directives to program indirect command files that control your use of the system. Indirect is described in full in the RSX-11M/M-PLUS Indirect Command Processor Manual.

Note, however that, if one indirect command file contains commands to both the MCR and DCL 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 to DCL afterwards.

### 1.6 DCL AND RSX-11M-PLUS UTILITIES AND SYSTEM TASKS

The RSX-11M-PLUS 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-PLUS Task Builder. Appendix A of this manual gives information on the relationships between DCL and the utilities and other system tasks.

For more information on each RSX-11M-PLUS 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.



## HOW TO USE DCL

### 1.7 RSX-11M-PLUS DCL AND VAX/VMS DCL

DCL on RSX-11M-PLUS Version 3.0 and subsequent releases is designed to be compatible with DIGITAL Command Language 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-PLUS system or a VAX/VMS system. 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-PLUS, will have system-specific extensions to DCL. For example, RSX-11M-PLUS systems have several features not available on VMS systems; DCL commands have been implemented to invoke these features. To avoid confusion, however, the command syntax for these features differs from VMS command syntax.

In addition, many of the features discussed in this chapter are RSX-11M-PLUS extensions to the basic DCL language. Prompts, HELP, error messages, system displays, and command parsing (including abbreviating) are unique to RSX-11M-PLUS 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 the interactive users at terminals. You should not use these brief command forms, or other abbreviations shorter than four characters for each 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.

#### 2.1 SET AND SHOW TIME

The format for the SET [DAY]TIME command is as follows:

##### Format

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

## THE SET AND SHOW COMMANDS

You can set both the time and the date with this privileged command. You can enter the date in either form, for example, 04-JUN-85, or 6/4/85. In the latter case, mn 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 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

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 hard copy 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 8.4 for more information on the memory display. See Section 7.14.3 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 SHOW TERMINAL command without any qualifier, displays the current setting of all settable attributes of the terminal.

### Format

SHOW TERMINAL

## THE SET AND SHOW COMMANDS

You use this command in the following manner:

\$ **SHOW TERMINAL** **(RET)**

```
TT13:  [TEST1]                07-MAY-859:05      1      M. ANGEL
      CLI   = DCL      BUF   = 80.      HFILL  = 0
      LINES = 24.      TERM  = VT2xx  OWNER  = none  BRO    NOABAUD
      LOWER NOPRIV    NOHOLD  NOSLAVE  NOESC   CRT    NOFORM  NOREMOTE
      ECHO   VFILL    HHT     NOFDX    WRAP    NORPA   NOEBC   TYPEAHEAD
      CTRLC  AVO     ANSI    DEC      EDIT    NOREGIS NOSOFT  NOBLKMOD
      SERIAL
```

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

Terminal number, login directory, date and time, and name

CLI	Current CLI setting
BUF	Buffer length, that is, the width in columns
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
CTRLC	Terminal has Control/C ( <b>CTRLC</b> ) abort enabled

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	/CONTROL=C	/HOLD SCREEN	/INQUIRE

You can also set your terminal as a particular model:

/ASR33	/LA36	/RT02C	/VT102
/ASR35	/LA38	/VT05	/VT105
/DTC01	/LA50	/VT50	/VT125
/KSR33	/LA100	/VT52	/VT131
/LA12	/LA120	/VT55	/VT132
/LA30P	/LA180S	/VT61	/VT200_SERIES
/LA30S	/LN03	/VT100	
/LA34	/RT02	/VT101	

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 [dd:]

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 5.13.3.

## 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.

This display includes batch queues as well as print queues.

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 Section 4.10.2.2 and the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information on the SHOW QUEUE commands.

### 2.5.1 SHOW PROCESSOR

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

### Format

SHOW processortype processorname[/qualifier[s]]

## THE SET AND SHOW COMMANDS

### Parameters

#### processortype

The processor type is one of the following:

BATCH  
CARD\_READER  
DEVICE  
INPUT  
PRINTER  
PROCESSOR

The CARD\_READER and INPUT arguments are synonyms. The DEVICE argument refers to all nonbatch output processors and is a synonym for PRINTER. The BATCH argument refers to all batch processors. The PROCESSOR argument refers to all processors.

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

See Section 4.10.2.3 the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information.

#### processorname

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

Batch processors have names in the form BAPnnn, where nnn is from one to three characters.

Applications processors have names of six characters.

### 2.6 SHOW ASSIGNMENTS AND SHOW LOGICALS

These commands display at your terminal all local and login logical-device assignments. See Section 5.13.1.

#### Format

SHOW ASSIGNMENTS[:ttnn:]

SHOW LOGICALS[:ttnn:]

### 2.7 SET AND SHOW DEFAULTS

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

#### Format

SET DEFAULT[/qualifier] [ddnn:][directory]

#### Command Qualifier

/[NO]NAMED\_DIRECTORY

## THE SET AND SHOW COMMANDS

### Parameters

dd

The 2-letter device mnemonic

nn

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

[directory]

The directory can be in two forms: named or numbered. A named directory is from one to nine of the following characters: the 26 letters A through Z, and the numbers 0 through 9. A numbered directory is two octal numbers from 1 through 377, 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 4.11.1.

The SHOW DEFAULT command displays your current default device and directory. See Section 4.11.2.

### Format

SHOW DEFAULT

## 2.8 SHOW TASKS

This command displays information about tasks in the system. Six forms of the command are most useful, as follows:

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

### 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 SET AND SHOW COMMANDS

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 two commands display full information on a single active or installed task. See Section 7.14.3.

### Format

SHOW TASKS/ACTIVE/FULL taskname

SHOW TASKS/INSTALLED/FULL taskname

## 2.9 SHOW USERS

On RSX-11M-PLUS systems without Resource Accounting, the SHOW USERS command displays the terminal number and login UIC for each logged-in terminal.

On RSX-11M-PLUS systems with Resource Accounting, this command also displays for each logged-in terminal the default directory, the date and time of logging in, the number of currently active tasks, and the user's name.

### Format

SHOW USERS

See Section 8.3 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)

### Parameter

filespec

The filespec uniquely identifies a file. The filespec 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.

## THE SET AND SHOW COMMANDS

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 the 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.

See Section 4.11.6 for more information.

You can display the protection of a file with the DIRECTORY command and /FULL qualifier. See Section 4.8.1.

### 2.11 SET AND SHOW SYSTEM/DIRECTORY AND LIBRARY/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 [1,54] on the system disk. An installation may, however, have more than one directory 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 AND SHOW COMMANDS

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

### Format

SET SYSTEM/DIRECTORY:[directory]

The SHOW SYSTEM/DIRECTORY command displays the current system directory as follows:

### Format

SHOW SYSTEM/DIRECTORY

On RSX-11M-PLUS systems, the utilities and other nonprivileged tasks are kept together in the library directory. In most cases, this will be the directory [3,54] on the system disk.

The SET LIBRARY/DIRECTORY privileged command establishes the library directory on RSX-11M-PLUS systems as follows:

### Format

SET LIBRARY/DIRECTORY:[directory]

The SHOW SYSTEM/DIRECTORY command displays the current library directory as follows:

### Format

SHOW LIBRARY/DIRECTORY

The system and library directories make possible the RUN \$taskname convention. When you issue a RUN or INSTALL command with the task name preceded by the dollar sign (\$), you are instructing the system to look first in the system directory and then in the library directory 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 task by device and directory.

See Section 8.2 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 the /NOEXECUTE qualifier. 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. Note that you should not use this command if your terminal is set to serial mode. See Section 1.3.15 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-PLUS operating system. For a quick review of basic operations at a terminal, refer to the Introduction to RSX-11M-PLUS.

The discussions in this chapter refer to three DIGITAL terminals: the LA36 hardcopy terminal and the 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. In general, what is true of the LA36 will be true of many models of hardcopy terminal, and what is true of the VT100 will be true of many models of video terminals.

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 sort of keyboard; some terminals will not have a keypad.

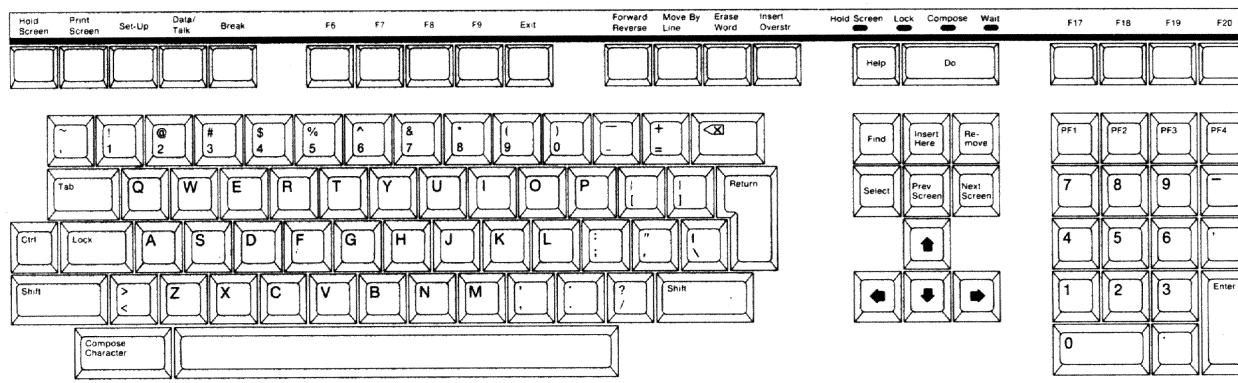


Figure 3-1 VT200-Series Keyboard and Keypad

Each terminal on a RSX-11M-PLUS 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 Chapter 7, 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 displays the terminal number and default directory for each logged-in user.

On systems with Resource Accounting active, SHOW USERS also displays the user's name and the number of currently active tasks.

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 displays 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 the dollar sign prompt (\$) to appear.

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, look at the prompt to determine whether the terminal is set to DCL or MCR. If the terminal is set to MCR, a right-angle-bracket prompt (>) is displayed on the terminal.

If the terminal is set to MCR, issue the following command line to determine if there are any active tasks running on the terminal:

>ACT 

## TERMINAL OPERATIONS

If the terminal is set to DCL, issue the following command for the same information:

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

This command informs 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:

```
MCR>BYE (RET)
```

Then, after you log out the terminal, log yourself in.

If the terminal is not logged in, you may log in. You can log in using either HELLO or LOGIN together with your User Identification Code (or name) and your password. The system's HELLO program logs you in if you have an account on the system and have used the proper password. 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 converse) with the following commands:

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

or

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

All RSX-11M-PLUS systems are multiuser protection systems. This means that the system includes features, such as LOGIN and LOGOUT (Section 3.1.2) and ALLOCATE or MOUNT/NOSHAREABLE( Chapter 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 the system. LOGIN also establishes certain characteristics of your terminal session.

### Format

```
LOG[IN]
userid
Password:  password

LOGIN  userid
PASSWORD:  password

LOGIN  userid/password
```

### Parameters

#### userid

Identifies the user logging in. It is easiest to log in by name. However, this command also accepts four forms of User Identification Code (UIC):

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

Each user has 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 using either a comma in the UIC---for example, [303,17]---or your name, a file called LB:[1,2]LOGIN.TXT normally prints on your terminal. Your system manager puts information about the system and other announcements in this file.

You can suppress LOGIN.TXT messages by using a slash (/) instead of a comma (,) between the group and member numbers of your UIC (for example, [303/17]) when you log in.

Your system manager can set up your account so that messages will not print on your terminal when you log in or log out. If your system manager selects this option for your account, you will not receive LOGIN.TXT messages regardless of how you log in.

#### password

Your password is limited to 39 Radix-50 characters, including dollar signs (\$), periods (.), exclamation points (!), quotation marks ('), and hyphens (-). When you enter your password in response to the Password: prompt, your password does not print on the screen. However, if you type in your password on the same line as your user identification, the password does print. Your system manager establishes your password as part of setting up your account. You can change your password with the SET PASSWORD command. See Section 3.1.3.1.



## LOGIN (Cont.)

### Examples

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

RSX-11M-PLUS V3.0 BL24 Multiuser [3,54] System CASTOR
20-MAY-85 09:16 Logged on Terminal TT54:

Good Morning

May 20, 1985          System Manager

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

      ** BACKUPS TONIGHT **

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

May 1, 1985          Operator

Magtape drive MS0: is back on line.
```

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

The dollar sign prompt (\$) signifies the completion of logging in.

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

RSX-11M-PLUS V3.0 BL24 Multiuser [3,54] System CASTOR
20-MAY-85 09:16 Logged on Terminal TT54:

Good Morning

May 20, 1985
System Manager

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

      ** BACKUPS TONIGHT **

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

## LOGIN (Cont.)

This example shows the first login of the day for this user. The UIC is entered with a slash (303/5). The printing of the system LOGIN.TXT file in full is therefore suppressed. Only the first message (defined as being a priority message by the system manager) is 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-PLUS V3.0 BL24 Multiuser [3,54] System CASTOR
20-MAY-85 09:16 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-PLUS V3.0 BL24 Multiuser [3,54] System CASTOR
20-MAY-85 09:16 Logged on Terminal TT54:
```

Good Evening

```
$ @LOGIN.CMD
$ SET TERMINAL/INQUIRE
$ SET PROTECTION:(S:RWED,O:RWED,G:R,W:)/DEFAULT
$ ASSIGN RS: DUL:
$ @ <EOF>
```

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

Your system manager may have established your account so that you do not receive login messages, regardless of how you log in. If this is the case, none of the information that usually appears between the Password: prompt and your LOGIN.CMD file prints on your terminal. Therefore, you do not see the system identification or messages from the system manager's LOGIN.TXT file,

## LOGIN (Cont.)

### Notes

HELLO is identical to LOGIN.

LOGOUT, or BYE, counteracts LOGIN.

When you log in, the system's HELLO task establishes many characteristics of the terminal session you are beginning. These include your default device (where your files are located), whether your account is privileged or not, and so forth.

Your system may have a file called LB:[1,2]SYSLOGIN.CMD. This file usually contains system-level commands that your system manager wants the operating system to execute each time a user logs in.

You can create a file called LOGIN.CMD. You keep this file in your directory, and the operating system executes it each time you log in. A LOGIN.CMD file usually includes commands for setting up your terminal and your account for regular use, although this file may contain any commands you want. See the examples. See also the Introduction to RSX-11M-PLUS and the RSX-11M/M-PLUS Indirect Command Processor Manual for more information on indirect command files.

It is wise to log in occasionally with either your name or a comma in your UIC. This way you can be sure you have not missed any important messages.

### 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 stored in the account file.

**User Action:** Retype the command using correct data.

## LOGIN (Cont.)

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 logged in terminal. Issue SHOW DEVICES TT: and find out which terminal has your terminal allocated.

## LOGOUT

### 3.1.2 LOGOUT

LOGOUT logs the user off the 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/NOSHAREABLE).

#### 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
Connect time: 0 hrs 44 mins 0 secs
CPU time used: 0 hrs 12 mins 6 secs
Task total: 90
Have a good afternoon
08-JUL-85 16:58 TT10: Logged off PRINCE
```

In this example, the user logs off an RSX-11M-PLUS system. The information on system use only prints on systems that have Resource Accounting running. (PRINCE is a DECnet node name. DECnet is a DIGITAL product that allows the users of different computer systems to share information and resources. A node is the name of one of these systems.)

Your system manager may have set up your account so that you do not receive system information when you log out. In this case, the operating system simply prints a dollar sign prompt (\$) to indicate that you have successfully logged out.

## LOGOUT (Cont.)

```
$ LOGOUT RET
DMO -- TT10: Dismounted from DU1:
Connect time:    0 hrs 32 mins 0 secs
CPU time used:  0 hrs    0 mins 12 secs
Task total:      21
Have a good morning
08-JUL-85 11:17 TT10:  Logged off ROMPER
```

In this example, the user logs off with a volume mounted on device DU1:. The LOGOUT command dismounts the volume, as the message states. If the device is allocated, the LOGOUT command deallocates it, but does not issue a message.

```
$ LOGOUT RET
13:29:36 Task "TT10 " Terminated
          Aborted via directive or CLI

Connect time:    0 hrs 23 mins 0 secs
CPU time used:   0 hrs  3 mins  6 secs
Task total:      36
Have a good afternoon
19-JUN-85 13:30 TT10:  Logged off MICRO
```

In this example, the user has a task running from the terminal at the time the LOGOUT command is issued. The LOGOUT command causes the task to be aborted.

### Notes

LOGIN, or HELLO, counteracts LOGOUT.

You need type only L0 to log out.

BYE is the same as LOGOUT.

### 3.1.3 ACNT, the Account File Maintenance Program

Account files on RSX-11M-PLUS 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 (commonly DCL) 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 have no access to ACNT. They can change their password using the SET PASSWORD command (the next section). However, to change other aspects of their account, nonprivileged users need to see the system manager.

The RSX-11M/M-PLUS System Management Guide also has information on Resource Accounting, a separate system program that keeps track of system usage. Resource Accounting is briefly described in Section 3.2.3.2.

## SET PASSWORD

### 3.1.3.1 SET PASSWORD

SET PASSWORD changes your password.

#### Format

SET PASSWORD

Old password:

New password:

Verification:

Nothing that you type in response to these prompts is echoed. This is shown in the examples by enclosing the password in angle brackets.

Your password is limited to 39 Radix-50 characters, including dollar signs (\$), periods, exclamation points (!), quotation marks and hyphens (-).

#### Example

```
$ SET PASSWORD   
Old password: <KRYPTO>   
New password: <COVERT>   
Verification: <COVERT> 
```

In this example, the user changes the password from KRYPTO to COVERT. SET PASSWORD verifies that the new password is correct. If you type in different responses to New password: and Verification:, the password is not changed.

To preserve security, choose nonobvious passwords. You should not use your name (or a close friend's), nor should you use the same password on more than one system. The more characters there are in a password, the harder it is to guess.

#### Notes

SET PASSWORD invokes a task called PSW, which is installed as ...PSW. PSW puts the new password in the system account file.

Because passwords are encrypted on RSX-11M-PLUS, the system manager and other privileged users cannot access your password, however, they can still change it by running ACNT.

If you are a nonprivileged user and do not remember your password, you will have to have the system manager or a qualified privileged user change your password by running ACNT.

## SET PASSWORD (Cont.)

### Error Messages

SET---ERROR, Task PSW is not installed

**Explanation:** The task that changes passwords is not installed.

**User Action:** Ask your system manager to install PSW or to change your password in the system account file.

SET---Illegal password syntax

**Explanation:** The password you typed in was illegal.

**User Action:** Type in a password consisting of 39 or fewer Radix-50 characters. (\$), periods (.), The Radix-50 characters are the 26 uppercase letters, the numbers zero through nine, the dot character (.), the dollar sign (\$), the exclamation point (!), the quotation mark ('), and the hyphen (-).

SET---New password verification error

**Explanation:** The password you typed in response to the **Verification:** prompt was not the same as the one you typed in response to the **New password:** prompt.

**User Action:** Nothing has happened. Your password is unchanged. Type the command again, carefully.

SET---Old password validation error---password unchanged

**Explanation:** You typed in your old password incorrectly.

**User Action:** Check for correct password and retype it carefully.

SET---Timeout on response

**Explanation:** You typed in the SET PASSWORD command, but allowed a prompt to wait 60 seconds without a response.

**User Action:** Nothing has happened. Your password is unchanged. Type the SET PASSWORD again and respond more rapidly.

### 3.1.3.2 Resource Accounting on RSX-11M-PLUS

RSX-11M-PLUS systems include Resource Accounting as a system-generation option that provides a file of information on system use. Data is generated both for the individual user and for the system.

When you log in, after LOGIN checks the validity of your UIC and password, Resource Accounting assigns a number to the terminal session. The number can optionally include a 1- to 3-character personal identifier. Thus, each time you log in, the terminal session is identified by the number and the personal identifier. If you are logged in on two terminals at once, each session has a different number but the same identifier.



## TERMINAL OPERATIONS

When you log out, Resource Accounting displays the time you were logged in, the CPU time used, and the number of tasks used.

Users can also display current Resource Accounting data with the command `SHOW ACCOUNTING/INFORMATION`.

The data gathered by Resource Accounting makes possible the billing of individual users for the system resources used, and measurement of overall system usage.

For further information on Resource Accounting as well as command descriptions, see the `RSX-11M/M-PLUS System Management Guide`.

### 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 (Section 3.2.7) 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.

#### 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.

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, - (RET)
->MADISON.AVE, FUZZY.TXT;12, GUNGA.DIN;2 (RET)
```

## TERMINAL OPERATIONS

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-JUN-85
```

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 moves the cursor to the beginning of the current line, 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, as it is in EDT.)

The **DELETE** key (labeled **RUBOUT** on some terminals and  on the VT200-series terminals) works in a slightly different fashion on video and hardcopy terminals.

## TERMINAL OPERATIONS

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 sequence:

```
$RUN FEEBER  DEL  DEL  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, thus:

```
$ RUN FEEBER\RE\LE  CTRL/R
$ 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  DEL  DEL  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.

### 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-11M-PLUS" 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 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

## TERMINAL OPERATIONS

you want 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 the VT100- and VT200-series terminals repeat automatically if you keep the key depressed. There is no **REPEAT** key on these terminals.

### 3.2.6 NO SCROLL or HOLD SCREEN

The **NO SCROLL** key on the VT100-series terminals and the **HOLD SCREEN** on the VT200-series terminals control the flow of information on your terminal screen. These keys are useful when you want to read text on your terminal screen more slowly than the terminal presents it. Press **NO SCROLL** or **HOLD SCREEN** to stop information from moving upward and off your screen; you can press either key again to resume the flow of information.

Note that you do not lose any information when you use the **NO SCROLL** or **HOLD SCREEN** keys.

See also the discussions of the **CTRL/S** and **CTRL/Q** keys in Section 3.4.2 and the **/[NO]TTSYNC** qualifier in Section 3.8.1.

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.

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 control key sometimes echoes as a circumflex when struck in combination with a letter.</p> <p>If you want to include <b>CTRL/a</b> characters in a batch job, the letter preceded by the circumflex is treated as a <b>CTRL/a</b>.</p>

(continued on next page)

# TERMINAL OPERATIONS

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

Character	Name	Comments
\$	Dollar sign	Standard DCL prompt. Must be set in the startup command file STARTUP.CMD, using the CLI command. It may be different on your system.
	Prompt	Signifies that system is ready to accept commands.
>	Right-angle bracket	General term.
	greater-than	ASCII name of characters. Used in mathematical and logical expressions.
	Prompt	Standard MCR prompt. It may be different for your system.
<	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.
		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.

(continued on next page)

## TERMINAL OPERATIONS

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

Character	Name	Comments
*	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.
=	Equals	Can also be used preceding argument in DCL command. Also used as logical and mathematical operator in expressions.
{ }	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.
\	Backslash	On hardcopy terminals, delimiter for deleted characters.

### 3.3 KEYPADS

The VT100-series 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 alternate keypad.

The keys of the alternate keypad are available for special functions. When you use EDT, you can use these keys to easily edit text. In addition, some functions of the operating system use these keys. They may be used in many other ways, depending on the other applications your operating system is running.

## TERMINAL OPERATIONS

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 can be used for including the decimal point in numbers.
ENTER	Works the same as the <b>RETURN</b> 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.

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 gray) 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 plus one or more other characters to the operating system. The **ESCAPE** is a nonprinting character, but the associated characters echo on the terminal and may make the operating system behave as if you had typed those characters.

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

Key	Result
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~

## TERMINAL OPERATIONS

### 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 (^). Enter a **CTRL/U** and you see a ^U on your terminal.

The operating system recognizes 13 letters combined with the **CTRL** key. The letters are C, G, I, K, L, M, 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.

#### 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** can have either of two functions on a RSX-11M-PLUS system.

If your system has **CTRL/C** abort support and you have it set up for your terminal, **CTRL/C** aborts the task or tasks running from your terminal. If you press **CTRL/C** and it does not do this, issue the command **SET TERMINAL/CONTROL=C**. See Section 3.8.1 for more information.

If you have issued the command **SET TERMINAL/NOCONTROL=C**, **CTRL/C** is a line terminator. Its main use is to gain the attention of the operating system. In this form, **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 like the following:

DCL>

You can then type a command without interrupting any other tasks active at your terminal. In this form, **CTRL/C** also assures you that the command you enter goes to DCL when the **DCL>** prompt is showing.

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



## TERMINAL OPERATIONS

This second form of CTRL/C has special effects:

- If 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.
- 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 type a CTRL/C than to type SET TERMINAL/NOHOLD\_SCREEN. It does not matter whether you type in a command in response to the explicit DCL prompt (DCL>); hold-screen mode is disabled anyway. Hold\_screen mode is available in this form on VT52s only.

### 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. They are particularly useful on video terminals where the output may go by too fast to be read, but they work 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 data to your terminal, CTRL/C has the same effect as CTRL/Q: typing CTRL/C also resumes the flow of information.

If you use these control characters, you do not skip over any of the output. All output that should appear, does 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 typed CTRL/S. Type 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 a default, you do not need to set this characteristic unless you have changed it previously to /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.

## TERMINAL OPERATIONS

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

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

the system starts with the first file of the type **.LST** in your directory. If you suppress this output with a **CTRL/O**, the system starts 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 does not respond, a **CTRL/Z** will often clear it.
- 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 appears 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.

## TERMINAL OPERATIONS


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 Chapter 1 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 canceled. 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   
$ THINK
```

You may then continue typing the line.

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 often 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 typeahead buffer. This buffer is used to store input characters temporarily until the system can process them.

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) or the "keyboard lock" light is on. When this happens, enter 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 buffer.

### 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 always adds several line feeds at the point at which it is inserted. On a line printer, a CTRL/L usually moves the paper past the next perforation.

## TERMINAL OPERATIONS

### 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 or, for a video terminal, performs four line feeds.

**CTRL/M** is the same as **RETURN**.

# HELP

## 3.5 HELP

HELP displays information about DCL 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
/DCL
/FILE:filespec
/filename
/MCR
```

### 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 dollar sign prompt (\$) and to the explicit DCL prompt (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 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 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 is 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. You must be logged in to use this qualifier. 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.

#### /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 %. You must be logged in to use this qualifier. Examples follow at the end of this section.

#### /GROUP

Specifies that the help text is in the file HELP.HLP in the directory. This directory has a name consisting of the current group for the first three numbers and 001 for the last three numbers. If the current group is 303, then this directory is called [303,001]. All users with the same group number have access to this file by typing HELP/GROUP. You must be logged in to use this qualifier. Examples follow at the end of this section.

#### /CLI:cliname

Specifies that the help text begins in the file LB:[001,002]cli.HLP. This qualifier is for installations with alternate CLIs for which HELP is provided. The cliname can range from one through six Radix-50 characters. Examples follow at the end of this section.

#### /DCL

Specifies that you want help on a DCL command. For this qualifier, the help text begins in the file LB:[001,002]DCL.HLP. This is the default for terminals set to DCL.

#### /MCR

Specifies that you want help on an MCR command. For this qualifier, the help text begins in the file LB:[001,002]MCR.HLP. This is the default for terminals set to MCR.

## HELP (Cont.)

/FILE:filespec

Specifies any file where help text is located. If you do not give a complete file specification, the defaults are LB:[001,002]filename.HLP. You must be logged in to use this qualifier.

/filename

Specifies that the help text begins with LB:[001,002]filename.HLP. When using this qualifier, you cannot specify the device, directory, or extension, only the file name. You must be logged in to use this qualifier.

### Examples

\$HELP 

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	DEFINE	DELETE	DIFFERENCES
DIRECTORY	DISMOUNT	EDIT	FIX	FORTTRAN
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 types HELP without parameters or qualifiers.

## HELP (Cont.)

```
$ 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
```

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

File(s)?

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.

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

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

ACCOUNTING	[DAY]TIME	LIBRARY	PROCESSOR	TASKS
ASSIGNMENTS	DEFAULT	LOGICALS	PROTECTION	TERMINAL
CACHE	DEVICES	MEMORY	QUEUE	UIC
CLOCK-QUEUE	ERROR LOG	PARTITIONS	SYSTEM	USERS
COMMON	GROUPFLAGS			

```
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?

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.



## TERMINAL OPERATIONS

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 one through nine, 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.

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 one through nine 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.

## TERMINAL OPERATIONS

Further help is available. Type the following:

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 the following:

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 the following:

HELP FIRST

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 the following:

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 the following:

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 file specification can be any legal file specification. 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.

## TERMINAL OPERATIONS

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 following 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 follows. 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.

```
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, lavender, heliotrope, 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 username message

BROADCAST @filespec
```

#### Command Qualifiers

```
/ALL
/LOGGED_IN
```

#### Parameters

ttnn:

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

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 11.8.

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.

username

Specifies the user to whom you want the message to be sent. The message goes to all terminals the user is logged in on.

If there are two or more users with the identical last names on the system and you want to send a message to only one of them, you can precede their last name with their first initial followed by a period (for example, R.ROBUST and Q.ROBUST).

## BROADCAST (Cont.)

### @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.

### Command Qualifiers

/ALL

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

/LOGGED IN

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

### Examples

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

The message is printed on TT20:

```
30-MAY-85 16:34          FROM:  MICRO::SYSMGR (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 enter 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)
```

## BROADCAST (Cont.)

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

```
$ BRO @DAILY ␣
```

The file DAILY.CMD contains the following messages:

```
TT1:WHERE IS THE DITKO CONTRACT?
TT3:MEET ME FOR LUNCH AT 11:30.
TT7:THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
TT5:GET BACK TO WORK.
TT4:I LOST MY GREEN EYESHAD. HAVE YOU SEEN IT?
```

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

### Notes

B is the short form of BROADCAST.

Only terminals can receive messages.

If you want 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 want 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.

## BROADCAST (Cont.)

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.

REQUEST

BRO---User not receiving messages

BRO---TTnn:message

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

**User Action:** Wait and try again.

Message failed to reach TTnn:

**Explanation:** The message sent did not arrive at the specified terminal.

**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 drive is down again.  (RET)
```

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

```

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

      12-JUN-85 13:32      FROM: TT6:      TO: CO:
      CHARLIE, THE TAPE DRIVE IS DOWN AGAIN

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

The user at TT6: observes a problem on the system and reports it to the operator with REQUEST. The message appears on the operator's console, timestamped 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 comes 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

Many of these qualifiers can have NO in front of them. For instance, you can set a terminal with the /BROADCAST qualifier (the default), or the /NOBROADCAST qualifier. The terminal attributes are grouped below according to function and listed in alphabetical order within functions.

#### Group 1: Common Use

/BROADCAST	/CLI=cliname
/CONTROL=C	/DCL
/HOLD SCREEN	/INQUIRE
/LOWERCASE	/MCR
/PRIVILEGED	/SPEED:(transmit,receive)
/UPPERCASE	/WIDTH:n

#### Group 2: Terminal Setup

/ASR33	/ASR35
/DTC01	
/KSR33	/KSR35
/LA12	/LA30P
/LA30S	/LA34
/LA36	/LA38
/LA50	/LA100
/LA120	/LA180S
/LA210	
/LN03	
/LQP02	/LQP03
/PRO SERIES	
/VT05B	/VT50
/VT52	/VT55
/VT62	/VT100
/VT101	/VT102
/VT105	/VT125

## SET TERMINAL (Cont.)

/VT131	/VT132
/VT200_SERIES	
/ADVANCED_VIDEO	/ANSI_CRT
/AUTOBAUD	/BLOCK_MODE
/CRFILL	/DEC CRT
/EDIT_MODE	/FORM_FEED
/HARDCOPY	/HOSTSYNC
/LFFILL	/MODEL:n
/PAGE_LENGTH:n	/PRINTER_PORT
/REGIS	/SCOPE
/SOFT CHARACTERS	/TAB
/TTSYNC	
/[NO]TRANSLATION_ROUTINE[:arg]	
	n
	logical

### Group 3: Task Setup

/ECHO	/EIGHT_BIT
/ESCAPE	/FULL_DUPLEX
/INTERACTIVE	/LOCAL
/PARITY:arg	/PASSALL
ODD	/PASTHRU
EVEN	/REMOTE
/SERIAL	/SLAVE
/TYPEAHEAD[:n]	/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. Because 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.

## SET TERMINAL (Cont.)

### /[NO]BROADCAST

Establishes whether you want to receive broadcast messages on your terminal. The default, /BROADCAST, is to receive broadcasts.

The /NOBROADCAST qualifier limits the messages that appear on your terminal. This command prevents messages from appearing on your terminal that another user sends with the BROADCAST command. The operating system notifies senders that you are not receiving messages. In addition, this command restricts messages from the system shutdown task when it is shutting down the operating system. Your terminal does not display any messages about the impending shutdown of the system except those issued in the last five minutes. However, your terminal still receives messages from other tasks.

### /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.

### /[NO]CONTROL=C

Establishes whether a CTRL/C causes an abort (/CONTROL=C) or causes an explicit DCL> prompt (/NOCONTROL=C).

### /DCL

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

### /[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.

The /NOHOLD\_SCREEN qualifier 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.

The /HOLD\_SCREEN qualifier 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 NO SCROLL key once. If you want to see another screenful, press the NO SCROLL key while holding down the SHIFT key. Data is being held back; it is not being lost.

You can also control output to the terminal with CTRL/S and CTRL/Q. See Section 3.4.2.

On terminals in the VT100-series or the VT200-series, you can use either the NO SCROLL or HOLD SCREEN keys, respectively. See Section 3.2.6 for more information.

## SET TERMINAL (Cont.)

### /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, VT62, 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 types 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 the 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 if you do not always log in to the same terminal.

### /LOWERCASE /NOUPPERCASE

Characters typed on the terminal in lowercase appear in lowercase. This is the default. Most system tasks 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 inadvertently canceled the /LOWERCASE attribute.

/MCR Sets terminal to MCR. The 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 system-programmer 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 can be set by 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. When you issue SET TERMINAL/SPEED, you must separate the two rates with a comma. See the examples.

## SET TERMINAL (Cont.)

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 in uppercase. The /UPPERCASE attribute is the default. /NOLOWERCASE is the same as /UPPERCASE.

/WIDTH:n

Sets the width of your terminal, that is, the length of a line. The value n can be from 16 through 255.

### Group 2: Terminal Setup

The qualifiers in this section 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 the preceding section.

For certain common terminal 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 either must set the 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 identify a terminal that the /INQUIRE qualifier does not recognize to the system.

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

ASR33	ASR35	DTC01
KSR33	KSR35	LA12
LA30P	LA30S	LA34
LA36	LA38	LA50
LA100	LA120	LA180S
LA210	LN03	LQP02
LQP03	VT05B	VT50
VT52	VT55	VT62
VT100	VT101	VT102
VT105	VT125	VT131
VT132	VT200_SERIES	PRO_SERIES

## SET TERMINAL (Cont.)

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 Teletype<sup>1</sup> 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 is 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.

Note that you can also set terminal 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.

The following qualifiers set particular attributes of your terminal. See your terminal manual for more information.

### /[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 word SET-UP blinks in setup mode, you have the advanced video option.

### /[NO]ANSI\_CRT

States whether the terminal supports a subset of the ANSI standard. Terminals with the /ANSI\_CRT attribute do not necessarily interpret 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 characters 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 block-mode transmission.

---

1. Teletype is a trademark of the Teletype Corporation.

## SET TERMINAL (Cont.)

`/CRFILL:n`

Sets the attribute used to accommodate the carriage-return process on hardcopy terminals. While the terminal is performing the carriage return, 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 zero through seven.

`/[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`. Form feeds and vertical tabs will be simulated by the system by outputting the equivalent number of line feeds for terminals set `/NOFORMFEED`.

`/HARDCOPY`  
`/NOSCOPE`

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

`/[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 yet ready to accept. Once the system is able to process the characters in the typeahead buffer, 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.



## SET TERMINAL (Cont.)

### /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 set up 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 177 -- 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 the SHOW TERMINAL/MODEL.

If you want 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 comments in TTDAT.MAC.

### 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.

### /PRINTER\_PORT

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

## SET TERMINAL (Cont.)

### /[NO]REGIS

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

### /SCOPE /NOHARDCOPY

States whether the terminal is a video terminal or not. The /NOHARDCOPY qualifier is the same as /SCOPE.

### /[NO]SOFT\_CHARACTERS

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]TRANSLATION\_ROUTINE[:arg] [TTnn:]                                   n                                   logical

Sets your terminal to the specified character translation routine. You can specify only one character translation routine for a terminal. If your terminal is already set to a routine and you specify another routine, the old routine is disabled and the new routine is enabled for your terminal. Privileged users can set any terminal to a character translation routine. The /NOTTRANSLATION\_ROUTINE qualifier disables your terminal from using the character translation routine currently set for the terminal.

The n argument specifies the number of the character translation routine, which is assigned when the routine is installed.

The logical argument specifies the logical name for the character translation routine number which is assigned when the routine is installed.

### /[NO]TTSYNC

Controls whether the operating system responds to CTRL/S or CTRL/Q when you type these control characters on your terminal. The default, /TTSYNC, is for 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 any information when you use CTRL/S and CTRL/Q.

Note that, if you set your terminal /NOTTSYNC, the NO SCROLL key on VT100-series terminals and the HOLD SCREEN key on 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 set the attributes when they attach the terminal. User tasks can also do this. This is 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 seven-bit ASCII.

The /EIGHT\_BIT qualifier allows the terminal to pass all 8 bits of the ASCII character. This attribute is used when your terminal is communicating with some device that sends eight-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.

The /ESCAPE qualifier 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 no longer terminates a line. This characteristic must be set (by either the user or a task) for a task to read an escape sequence.

## SET TERMINAL (Cont.)

/[NO]FULL\_DUPLEX

Enables (or disables) full-duplex mode at a terminal.

The /NOFULL\_DUPLEX qualifier is the default. The terminal cannot process input and transmit output at the same time. "Half-duplex" is the usual industry term for this attribute.

The /FULL\_DUPLEX qualifier allows the terminal to process input and transmit 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. Special characters (such as CTRL/C) are interpreted by the system and cause special processing to occur (for example, gaining the attention of the system). 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 and the terminal.

The state of the parity bit, either 0 or 1, is determined by the argument used with the command. If SET TERM/PARITY:EVEN is specified, the state of the parity bit is selected to force the total number of "1" bits 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- and VT200-series terminals, this is done through the terminal's SET-UP mode.

## SET TERMINAL (Cont.)

SET TERM/PARITY is not related to /EIGHT\_BIT. 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.

/PASSALL  
/NOINTERACTIVE

Specifies that characters sent by the terminal should not be interpreted by the terminal driver. Therefore, the task must interpret all characters. For example, CTRL/C is passed directly to the task, which causes special system processing. This parameter automatically sets the terminal to the /EIGHT\_BIT attribute.

/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 qualifier 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]PASTHRU

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

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

/PASTHRU 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 /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 primarily used by applications programmers.

/[NO]SERIAL

Determines whether you have only one task running at your terminal, or many tasks running at your terminal. The default /NOSERIAL, on RSX-11M-PLUS means that when you enter a command before the previous command has completed execution, the second command is processed in parallel with the first.

## SET TERMINAL (Cont.)

When a terminal is set /SERIAL, commands are retained in the terminal's typeahead buffer until it is time for them to be passed to the system. See the discussion of /TYPEAHEAD for information on how to change the size of the terminal's typeahead buffer.

SET TERMINAL/SERIAL allows programmers to "stack" a compiler command, a LINK command, and a RUN command, allowing each command to use the output of the previous command. It is a simple way of executing a group of commands in order.

Using CTRL/C to abort tasks works unambiguously in the /SERIAL mode. One task is running; one is aborted.

On the other hand, it is often convenient to have more than one task running at your terminal. You may want to edit a file while also running a compiler or awaiting completion of a LINK command. In that case, you can set your terminal /NOSERIAL. Each command is executed as it is entered. Commands are executed in parallel. If you are set /NOSERIAL, watch the return of the dollar sign prompt (\$) carefully. If you are running two tasks, you get one dollar sign prompt (\$) per task when execution is completed. If you enter a compiler command followed immediately by a LINK command naming the compiler output, the LINK command fails.

When you are set /NOSERIAL, CTRL/C aborts affect all tasks that include your terminal number in their name. (See Chapter 7 for more information on task naming.) You may want to set your terminal /NOCONTROL=C. This permits you to abort tasks and commands by name.

### /[NO]SLAVE

Enables (or disables) slave status at a terminal.

/NOSLAVE 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 attach the terminal without specific action by the user.

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

### /[NO]TYPEAHEAD[:n]

Enables or disables the terminal's typeahead attribute and also sets the size of the typeahead buffer. /TYPEAHEAD:86 is the default for instruction and data-space systems. The typeahead buffer is used by the system to store characters received from the terminal before they are processed. 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.

## SET TERMINAL (Cont.)

On systems that support separate instruction and data space, you can set the size of the typeahead buffer from 0 through 255. If your terminal is set /SERIAL, you may want to increase the size of the typeahead buffer so you can queue more commands.

On non-instruction and data systems, the size of the typeahead buffer is fixed at 58 characters.

/[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 is the equivalent of the previous example. Include the command in this form in your LOGIN.CMD file and your terminal attributes will be set each time you log in.

```
$ 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 is the reprinting of characters deleted by the DELETE key. This command sets a number of defaults. These are shown in Table 3-2 in Section 3.8.2.

```
$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, you must reset the hardware to the same transmit and receive rates. You must issue the SET TERMINAL command first and then set the terminal, using SET-UP, so that the terminal driver can communicate with the terminal.

## SET TERMINAL (Cont.)

```
$ 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.

```
$ SET TERMINAL/SERIAL/TYEAHEAD:150 (RET)
$ MACRO/SHOW:ALL PETE, JOHN, ROGER, KEITH (RET)
$ LINK/TASK:WHO KEITH (RET)
$ RUN WHO (RET)
```

In this example, the user sets the terminal /SERIAL and increases the size of the typeahead buffer. The user then enters a MACRO command, followed by a LINK command and a RUN command. The commands are separated by RETURNS. The LINK command is not issued until the assembly completes, and the RUN command is not issued until the task build completes. In fact, the LINK command is not even echoed until the MACRO command completes, and the same is true for the RUN command. The unexecuted commands are stored in the typeahead buffer.

### NOTE

You can find how all attributes are set for your terminal with the SHOW TERMINAL command. See the next section.



# 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	/CLI:cliiname
/CONTROL=C	/CRFILL
/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	/PRINTER_PORT
/PRIVILEGED	/REGIS
/REMOTE	/SCOPE
/SERIAL	/SLAVE
/SOFT_CHARACTERS	/SPEED
/TAB	/TTSYNC
/TYPEAHEAD	/UPPERCASE
/WRAP	

### Terminal Types

```
/HT
/TI
/VT
```

### Defaults

```
/RT
/TT
```

### Terminal Models

/ASR33	/ASR35
/DTC01	/KSR33
/KSR35	/LA12
/LA30P	/LA30S
/LA34	/LA36
/LA38	/LA50
/LA100	/LA120
/LA180S	/LA210
/LN03/LQP02	
/LQP03	/PRO_SERIES

## SHOW TERMINAL (Cont.)

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

### Command Option

TTnn:

Identifies the terminal about which you want 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:TT6: (RET)
```

displays all the attributes 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 issue both a SHOW TERMINAL/BROADCAST command and a SHOW TERMINAL/NOBROADCAST command.

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 video terminal.

# SHOW TERMINAL (Cont.)

The same display results from the following:

**\$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.

/HT  
/RT  
/TI  
/TT  
/VT

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. /VT displays a list of virtual terminals. Virtual terminals are used in batch jobs and may be used by some applications.

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
CLI:cliname	CLI=	CLI:cliname
CONTROL=C	CTRLC	CONTROL=C
CRFILL	HFILL=	CRFILL
DEC CRT	DEC	DEC CRT
DCL	CLI=DCL	DCL
ECHO	ECHO	ECHO
EDIT MODE	EDIT	EDIT MODE
EIGHT BIT	EBC	EIGHT BIT

(continued on next page)

# SHOW TERMINAL (Cont.)

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

SHOW TERMINAL Attribute	SHOW TERMINAL Display	SET TERMINAL Attribute
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
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
PRIVILEGE	PRIV=	PRIVILEGE
PRINTER_PORT		
REGIS	REGIS	REGIS
REMOTE	REMOTE	REMOTE NOLOCAL
SCOPE	CRT	SCOPE NOHARDCOPY
SERIAL	SERIAL	SERIAL
SLAVE	SLAVE=	SLAVE

(continued on next page)

## SHOW TERMINAL (Cont.)

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

SHOW TERMINAL Attribute	SHOW TERMINAL Display	SET TERMINAL Attribute
SOFT_CHARACTERS	SOFT	SOFT
SPEED	SPEED=	SPEED
TAB	HHT	TAB
TTSYNC	TTSYNC	TTSYNC
TYPEAHEAD	TYPEAHEAD	TYPEAHEAD
UPPERCASE	NOLOWER	UPPERCASE
WIDTH	BUF=	WIDTH
WRAP	WRAP	WRAP

### Examples

```
$ SHOW RET
Function? TERMINAL RET
TT10: [DDALE] 16-MAY-85 13:26 1 D. DALE
      CLI = DCL      BUF = 80.      HFILL = 0      SPEED= (9600:9600)
      LINES = 24.    TERM = VT100    OWNER = none    BRO NOABAUD
      LOWER NOPRIV   NOHOLD NOSLAVE NOESC CRT NOFORM NOREMOTE
      ECHO NOVFill   HHT NOFDX WRAP NORPA NOEBC TYPEAHEAD
      CTRLC AVO      ANSI DEC NOEDIT NOREGIS NOSOFT NOBLKMOD
      SERIAL NOHSYNC NOPASTHRU TTSYNC
```

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 date and time, the number of active tasks, the name, plus all the attributes that can be set with SET TERMINAL. See Table 3-2 and the remaining examples.

```
$ SHOW TERMINAL:TT5 RET
TT5: [007100] 16-JUN-8513:47 1 B. DIDDLEY
      CLI = DCL      BUF = 80.      HFILL = 0      SPEED=(9600:9600)
      LINES = 24.    TERM = VT100    OWNER = SELF    BRO NOABAUD
      LOWER NOPRIV   NOHOLD NOSLAVE NOESC CRT NOFORM NOREMOTE
      ECHO NOVFill   NOHHT NOFDX WRAP NORPA NOEBC TYPEAHEAD
      CTRLC AVO      ANSI DEC NOEDIT NOREGIS NOSOFT NOBLKMOD
      SERIAL 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 (Cont.)

```
$ SHO TERMINAL/LA36/DCL (RET)
DCL=TT5
LA36=CO0:
LA36=TT0:
LA36=TT4:
```

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.

```
$ 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=TT2:
```

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=TT7:
```

```
•
•
•
```

```
$ SHO TERM/NOHARDCOPY (RET)
CRT=TT2:
```

```
•
•
•
```

```
$ SHO TERM/SCOPE (RET)
CRT=TT1:
```

```
•
•
•
```

```
$ SHO TERM/NOSCOPE (RET)
NOCRT=TT7:
```

CRT means cathode ray tube, another name for a video screen or scope.

```
$ SHO TERM/CRFILL (RET)
HFILL=TT10:0
```

This example returns the /CRFILL value for the terminal from which was issued.

## SHOW TERMINAL (Cont.)

```
$SHO TERM/LFFILL (RET)
VFFILL=TT4:
.
.
.
```

This example shows which terminal has its /LFFILL attribute enabled.

```
$ SHO TERM/FORMFEED (RET)
FORMFEED=TT3:
.
.
.
```

```
$ SHO TERM/NOFORMFEED (RET)
NOFORMFEED=TT5:
```

This example shows which terminals have their FORMFEED attribute enabled (or not).

```
$ SHO TERM/TAB (RET)
HHT=TT1:
.
.
.
```

```
$ 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 SHOW TERMINAL command, giving the name of the other terminal.)

```
$ SHO TERM/ECHO (RET)
ECHO=TT1:
.
.
.
```

```
$ SHO TERM/NOECHO (RET)
NOECHO=TT1:
```

```
$ SHO TERM/ESCAPE (RET)
ESCSEQ=TT10:
```

```
$ SHO TERM/NOESCAPE (RET)
NOESCSEQ=TT1:
.
.
.
```

ESCSEQ means ESCape SEquences.

## SHOW TERMINAL (Cont.)

```
$ SHO TERM/FULL_DUPLEX (RET)
FDX=TT2:
```

```
$ SHO TERM/NOFULL_DUPLEX (RET)
NOFDX=TT1:
.
.
.
```

FDX means full-duplex.

```
$ SHO TERM/PASSALL (RET)
RPA=TT1:
.
.
.
```

```
$ SHOW 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:
```

```
$ SHO TERM/LOCAL (RET)
NOREMOTE=TT2:
.
.
.
```

```
$ SHO TERM/NOLOCAL (RET)
REMOTE=TT1:
```



## SHOW TERMINAL (Cont.)

```
$ SHO TERM/NOREMOTE (RET)
NOREMOTE=TT2:
.
```

This attribute has no relationship to 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 no relationship to 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=TT5:
```

```
$ SHO TERM/NOSLAVE (RET)
NOSLAVE=TT1:
.
```

```
$ SHO TERM/TYPEAHEAD (RET)
TYPEAHEAD=TT1:
.
```

```
$ SHO TERM/NOTYPEAHEAD (RET)
NOTYPEAHEAD=TT7:
```

```
$ SHO TERM/WRAP (RET)
WRAP=TT1:
.
```

```
$ SHO TERM/NOWRAP (RET)
NOWRAP=TT4:
```

```
$ SHO TERM/NOEIGHT_BIT (RET)
NOEBC=TT1:
.
```

```
$ SHO TERM/EIGHT_BIT (RET)
EBC=TT6:
```

EBC means Eight-Bit Characters.



## CHAPTER 4

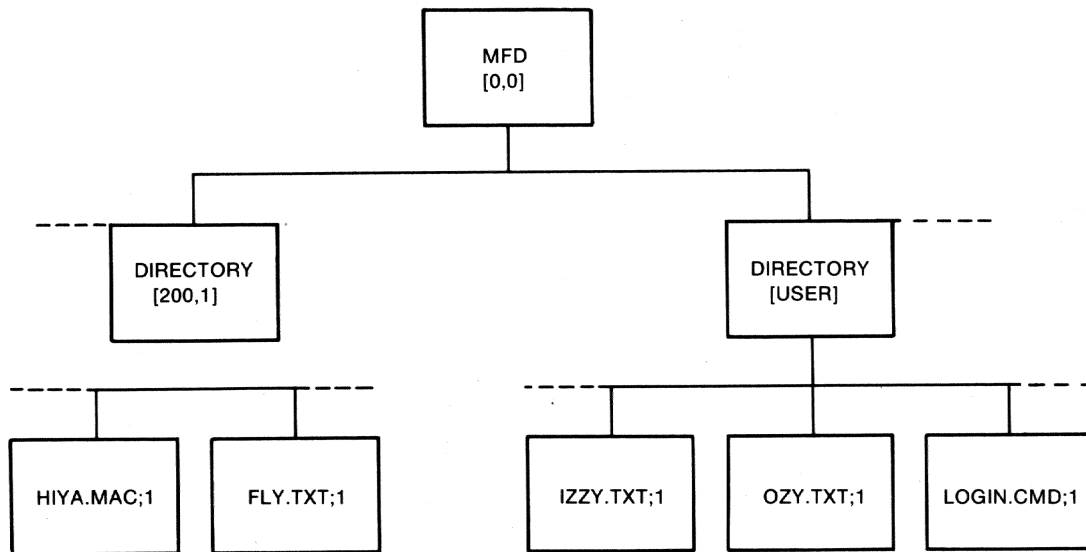
### HANDLING FILES

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 Chapter 10 of this manual. Other error messages are documented in the RMS-11 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 you mount the volume on the physical device, you can access the files on the volume. For more information about volumes and devices, see Chapter 5.

Within each disk 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 listed in the MFD are all User File Directories (UFDs). In general, UFDs are simply called directories.

See Figure 4-1 for an illustration of the relationship between volume, MFD, and directory.



ZK-3079-84

Figure 4-1 File Organization on a Volume

## HANDLING FILES

### 4.1 FILE OWNERSHIP AND LOCATION

When you log in, you identify yourself with a User Identification Code (UIC). You log in to your default device and directory. The default directory has a name (or number) assigned by the system manager when the account is established. Nonprivileged users can change their default device and directory, but not their UIC. Privileged users can change both. The UIC identifies the protection code for 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 protection UIC. The SHOW UIC command gives you your protection UIC. Both these commands are described in Section 4.11.2.

The default device is actually a volume mounted on a device. Your default device can always be called either by its name or by the name SY:. This is a logical assignment made each time you log in. Information on your default device is kept by the Account File Maintenance Program (ACNT), and picked up from there each time you log in. See the RSX-11M/M-PLUS System Management Guide for more information on ACNT.

When you create a file, the system places the file name in a directory along with a UIC. The UIC indicates the ownership of the file.

Each directory is a file with a name based on the name of the directory and the type .DIR. Thus, directory [KIZZY] is a file listed in the Master File Directory (MFD), and named KIZZY.DIR. Directory [2,1] is a file listed in the MFD, and named 002001.DIR.

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.

### 4.2 FILE SPECIFICATION

A file specification uniquely identifies a file, indicating its location and its contents. Many DCL commands require file specifications.

#### Format

ddnn:[directory]name.typ;ver

#### Parameters

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-PLUS are listed in Chapter 5.

## HANDLING FILES

### [directory]

Specifies the name of the directory in which the files are located.

RSX-11M-PLUS accepts either named directories or numbered directories. The format for a named directory is:

[directory] where directory is from one to nine of the following characters: the 26 letters A through Z, and the numbers 0 through 9.

Examples of named directories include [MINGUS], [RITTENBRG], [001002], and [A1B2C3].

The format for a numbered directory is:

[g,m] where g and m are octal numbers from 1 through 377.

Examples of numbered directories include [1,2] and [303,17].

See Section 4.11.1 for a description of the SET DEFAULT command. When your terminal's default is /NONAMED\_DIRECTORY, then you can specify only numbered directories in the SET DEFAULT command.

### name

Specifies the name of the file, which can be 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, the file type indicates the file contents. For example, .FTN is the file type for FORTRAN source programs.

Null file types (zero length) are acceptable.

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 4.3. Null file types (zero length) are acceptable.

### ;ver

Specifies the version number. If your system includes support for decimal version numbers, the version is a decimal number from 1 through 32767. If your system does not include decimal version number support, the version is 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.

## HANDLING FILES

You can also create a file with an explicit version number.

If you create a file with version number 32767, you can be sure that file cannot be inadvertently superseded by a file with a higher number; 32767 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-PLUS systems do not require you to enter the full file specification. Defaults are supplied for all fields of the file specification except the name. In addition, you can use wildcards to specify groups of related files (Section 4.4).

### 4.3 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:	The system establishes your default device 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 (Section 4.11.2). Remember that you do not have to use the unit number for devices numbered 0. DU: is the same as DU0:, for instance.
[directory]	The system manager specifies your default directory when establishing your account. Your default directory can be any legal directory name. Normally, your default directory is either your last name or a directory whose numbers are identical to your User Identification Code. You can change the default directory with SET DEFAULT (Section 4.11.1); you can display the default directory with SHOW DEFAULT (Section 4.11.2).
.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, that your default device is DU0:, which the operating system assigns the logical device name SY0:, and your default directory is [MUSTARD]. Assume further that there are three sequentially numbered versions of a file called TEXT.TXT in the

## HANDLING FILES

directory. If you want to print the highest numbered version of this file (version 3) on your terminal, all of the following forms of the TYPEcommand produce the same desired result:

```
$ TYPE TEXT.TXT
$ TYPE TEXT.TXT;3
$ TYPE TEXT.TXT;0
$ TYPE [200,20]TEXT.TXT;3
$ TYPE [MUSTARD]TEXT.TXT;3
$ TYPE DU0:TEXT.TXT
$ TYPE DU0:[200,20]TEXT.TXT
$ TYPE DU0:[MUSTARD]TEXT.TXT
$ TYPE SY:TEXT.TXT
$ TYPE SY:[200,20]TEXT.TXT;3
$ TYPE SY:[MUSTARD]TEXT.TXT;3
$ TYPE SY:[MUSTARD]TEXT.TXT;0
```

In most cases, you will use the first form, but if you want 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 is always assumed to be the highest numbered version, which is number 3 by definition in this example.)

Table 4-1  
File Types

File Type	Use
.BAS	BASIC-11 source program. System default.
.BAT	File containing batch processing commands. System convention.
.BLD	Indirect command file used to create files needed to build system tasks. System default.
.B2S	BASIC-PLUS-2 source program. System default.
.CBL	COBOL source program. System default.
.CMD	Indirect command file. System default.
.COR	Source Language Input Program (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-77 source program. System default.
.LOG	Log of batch processing session. System default.
.LST	Listing file. System default.
.MAC	MACRO-11 source program. System default.

(continued on next page)

## HANDLING FILES

Table 4-1 (Cont.)  
File Types

File Type	Use
.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 file. 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 the <u>RSX-11M/M-PLUS Utilities Manual</u> for more information on file types .PAT and .POB.
.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 Chapter 6, 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.

### 4.4 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 fields in the file specification, except the device name.

For instance, the following command:

```
$ DIRECTORY [*] 
```

lists all files on the default volume, regardless of the directory name.



The following command:

```
$ DIRECTORY [100,*] (RET)
```

lists all files in any numbered directories that have a group number of 100.

The following command:

```
$ DIRECTORY [*]TEXT.TXT (RET)
```

lists the most recent versions of all files on the default volume named TEXT.TXT, regardless of the directory name.

Likewise, the asterisk (\*) in place of the version number means "all versions."

The following command:

```
$ DIRECTORY WOM.BAT;* (RET)
```

lists all versions of the file WOM.BAT 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 *.BAT (RET)
```

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

The following command:

```
$ DIRECTORY COMMON.* (RET)
```

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 simplest 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, as does the SUBMIT command for batch processing.

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 more characters in this position."

Therefore, the following command:

```
$ DIRECTORY L*.TXT (RET)
```

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

## HANDLING FILES

The following command:

```
$DIRECTORY *L*.TXT (RET)
```

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

The same substitutions can also be used in file types, so that the following command:

```
$ DIRECTORY SNOBLO.L* (RET)
```

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 directory.

Note that you can use more than one wildcard in file names and file types.

The following command

```
$ DIRECTORY *F*D*.TXT (RET)
```

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, the same 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 (RET)
```

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%81.TXT (RET)
```

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

The wildcards can be combined in a single file specification.

The following command:

```
$ DIRECTORY %L*T.TM% (RET)
```

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.

#### 4.5 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 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 same 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 any part of the filespec argument to /EXCLUDE, but you cannot specify just wildcards in the file specification.

#### 4.6 PROTECTION

RSX-11M-PLUS is a multiuser system. Each user is able to work without interference from any other user. One way RSX-11M-PLUS provides data privacy and system security is through restrictions on access to volumes and the files on the volumes.

The best protection for sensitive files is to put them on a private disk, diskette, or tape, and then lock it away. The following section explains how files and volumes can be protected by software on RSX-11M-PLUS systems. No protection is completely safe, just as no lock is a defense against a determined and resourceful burglar.

In order to access a file, you must have access to the volume, access to the directory, and access to the file. Protection can be specified for all these entities.

## 4.6.1 File Protection

Each user has a unique User Identification Code--commonly called a UIC--that the system manager assigns while setting up your account. Your UIC identifies you to the system.

UICs look like this:

[303,5]

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

[g,m]

or

[uic]

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

When you log in with a number, that number is your UIC. If you log in with a name, there is still a UIC associated with the name. You can display your UIC with the SHOW UIC command.

The UIC itself indicates whether or not you are privileged. Privileged users have group numbers from 1 through 10. Nonprivileged users have group numbers from 11 through 377. Note that, if you are a privileged user, you can change your UIC. See Section 4.11.4 for a description of the SET UIC command.

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

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

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

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

## HANDLING FILES

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

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

Ordinarily, the operating system assigns the system's default protection code to files. In a protection code, the four types of file access are designated by their first letters. The default protection is as follows:

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

Note that under this default protection scheme, any user can read your files because the WORLD category has READ access. Furthermore, everyone in the GROUP category and all privileged users have full READ, WRITE, EXTEND, and DELETE access. They can all delete your files.

If you have files that you particularly want to protect against deletion, or if you want to limit the access people have to your files, use the SET PROTECTION or SET PROTECTION/DEFAULT commands, described in Sections 4.11.6 and 4.11.7. Be aware, though, that there is no way to protect your files from privileged users, because they can change the protection code for any file.

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

```
[RWED, RWED, RWED, R]
```

In the DIRECTORY display, the groups are not given, but are in the same order as before, SYSTEM, OWNER, GROUP, WORLD. That is, in the previous example, WORLD has READ privileges.

A number of DCL commands include qualifiers that help you use file protection. The most important is the /OWN qualifier to the COPY command. If you don't specify otherwise, when you copy one of your files to send it to someone else, you are the owner of the copy, even if it isn't in your directory. Assume you have UIC [303,5] and you are placing a copy of a file you own in a directory called [MOONDOG], which is owned by UIC [200,2]. You are using the following command:

```
$ COPY RAMP.TXT [MOONDOG]RAMP.TXT 
```

## HANDLING FILES

The file RAMP.TXT is in both directories, but its owner in both cases is [303,5]. The owner of [MOONDOG] has only WORLD READ access to their copy of RAMP.TXT, as seen in this directory listing, where the file owner is the first item on the second line of each file listing:

```
$ DIRECTORY/FULL RAMP.TXT, [MOONDOG]RAMP.TXT (RET)
```

```
Directory DU0:[OTHER]  
3-JUN-85 15:49
```

```
RAMP.TXT;52          (7516,5)      2./2.      23-APR-85 13:12  
[303,5] [RWED,RWED,RWED,R]
```

Total of 2./2. blocks in 1. file

```
Directory DU0:[MOONDOG]  
3-JUN-85 15:49
```

```
RAMP.TXT;1           (14343,6)     2./2.      3-JUN-85 15:48  
[303,5] [RWED, RWED, RWED, R]
```

Total of 2./2. blocks in 1. file

Grand total of 4./4. blocks in 2. files in 2. directories

The /OWN qualifier to the COPY command specifies that the recipient of the copy is the owner, not the sender. The following commands illustrate this:

```
$ COPY/OWN RAMP.TXT [MOONDOG]RAMP.TXT (RET)  
$ DIRECTORY/FULL RAMP.TXT, [MOONDOG]RAMP.TXT (RET)
```

```
Directory DU0:[OTHER]  
3-JUN-85 15:50
```

```
RAMP.TXT;52          (7516,5)      2./2.      23-APR-85 13:12  
[303,5] [RWED,RWED,RWED,RI]
```

Total of 2./2. blocks in 1. file

```
Directory DU0:[MOONDOG]  
3-JUN-85 15:50
```

```
RAMP.TXT;2           (14343,6)     2./2.      3-JUN-85 15:49  
[200,2] [RWED,RWED,RWED,R]
```

Total of 2./2. blocks in 1. file

Grand total of 4./4. blocks in 2. files in 2. directories

You do not need the /OWN qualifier if you are copying a file from somewhere else. Generally, when you create a file, you own it. When you copy a file to some other place, you are creating the file, so you must override your default ownership with the /OWN qualifier.

#### 4.6.2 Directory Protection

Because a directory is a file, it has a protection code like that of 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.

The default access for directory and file protection is the protection code the disk was mounted with. The default for mount protection is the protection code the volume was initialized with. See the discussion of CREATE/DIRECTORY in Section 4.7.2 and also Chapter 5 for more information.

The CREATE/DIRECTORY command includes a /PROTECTION qualifier that provides a means of overriding the default system protection. This qualifier sets the protection for the directory file itself, not for files in the directory.

The following example shows how to set protection on a new directory so that no one else can even list your files:

```
$ CREATE/DIRECTORY/PROTECTION:(SYSTEM:,OWNER:RWED,GROUP:,WORLD:) (RET)
Device and UFD?DU1:[RAGBAG] (RET)
```

This example creates a directory called [RAGBAG] on the volume mounted on device DU1:. The protection code is the protection code applied to the actual directory file DU1:[000000]RAGBAG.DIR. The protection code specifies that the owner (the person issuing the command) has full access to the directory and the files in it, and that SYSTEM, GROUP, and WORLD have no access. These three groups may not list the directory (no READ privilege), nor can they create files in the directory (no WRITE and EXTEND privilege), or delete the directory (no DELETE privilege). That is, only the owner can do anything with the directory.

If the directory [RAGBAG] already exists, you can limit access by resetting the protection of the directory file with the following command:

```
$ SET PROTECTION:(SYSTEM:,OWNER:RWED,GROUP:,WORLD:) [000000]RAGBAG.DIR (RET)
```

Remember that this command does nothing to the protection of the individual files within the directory. If a sophisticated user knows the file ID numbers of your files, they're still accessible. Remember also that although you can protect your files against SYSTEM access by privileged users, a privileged user can still issue a SET PROTECTION command and change the protection to SYSTEM:RWED, whereupon the privileged user has complete access.

You can display the protection of a directory file in the same way as any other file. The following command displays the protection for the directory created in the previous example:

```
$ DIRECTORY/FULL DU1:[000000]RAGBAG.DIR (RET)
```

#### 4.6.3 Volume Protection

A disk with files in RSX-11M-RLUS format on it is called a Files-11 volume. Volume protection works in much the same way as file protection except it affects the entire volume.

## HANDLING FILES

You can specify the volume protection at the time you initialize the volume or at the time you mount it. Both the INITIALIZE command and the MOUNT command (see Chapter 5) include a /PROTECTION qualifier and an /OWNER qualifier for setting volume protection. In addition, both MOUNT and INITIALIZE include a /FILE\_PROTECTION qualifier that permits you to specify the protection for any new files created on that disk volume.

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 as follows:

SYSTEM:RWCD,OWNER:RWCD,GROUP:RWCD,WORLD:RWCD

(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.)

If you need to know the protection of a disk volume (and the default file protection for new files created on the volume), include the /SHOW qualifier in your MOUNT command. See Chapter 5 for more information.



## CREATE

### 4.7 CREATING DIRECTORIES AND SEQUENTIAL FILES

The CREATE command without a qualifier enables you to create a sequential file directory at your terminal. The CREATE/DIRECTORY command permits a user to create a directory.

#### 4.7.1 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, press a CTRL/Z to close the file.

If you want 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.'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)
```

## CREATE (Cont.)

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) has been entered.

```
$ 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 (RET)
office until Friday afternoon. (CTRL/Z)
```

In this example, the user used CREATE to take notes on a phone call.

```
$ CREATE CHURCHYLA.FEM (RET)
(CTRL/Z)
```

In this example, the user used CREATE to create an empty file for later use.

### Notes

The function of CREATE differs slightly in batch jobs. See the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information.

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 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****4.7.2 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:][[directory]]

CREATE/DIRECTORY[/qualifier[s]] [ddnn:][[directory]]
```

**Command Qualifiers**

```
/ALLOCATION:n
/LABEL:volumelabel
/NOWARNINGS
/OWNER UIC:[uic]
/PROTECTION:(code)
```

**Parameters**

[ddnn:][[directory]]

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 want to create the directory is mounted. This parameter defaults to SY0:.

The volume must be a mounted, Files-11 volume.

**[directory]**

Specifies the name of the directory you want to create. This parameter defaults to the current directory.

Directory names are from one to nine of the following characters: the 26 letters A through Z, and the numbers 0 through 9.

To retain compatibility with other RSX systems, RSX-11M-PLUS also accepts numbered directory names. The format for a numbered directory is [g,m], where g and m are octal numbers from 1 through 377.

## CREATE/DIRECTORY (Cont.)

### Command Qualifiers

/ALLOCATION:n

Specifies the number of directory entries (file names) for which space is to be initially allocated. The value 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 directory can be allocated. If they do not match, the command is rejected.

If you do not specify this qualifier, the volume label is not checked.

/NOWARNINGS

Suppresses error messages resulting from the command.

/OWNER\_UIC: [uic]

Specifies the User Identification Code (UIC) of the directory's owner. A UIC establishes the protection status of the directory. See Section 4.6.2 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 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.

The default is the protection code the disk was mounted with. The default for mount protection is the protection code at initialization.

The default system protection code is as follows:

/PROTECTION:(SYSTEM:RWED,OWNER:RWED,GROUP:RWED, WORLD:R)

See also Section 4.6 for more information on file and volume protection.

### Examples

```
$ CREATE/DIRECTORY (RET)
Device and UFD? DU2:[REGIS] (RET)
```

**CREATE/DIRECTORY (Cont.)**

This example creates directory [REGIS] on the volume mounted on device DU2:. 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 DU2:[REGIS] (RET)
```

This example is identical to the previous one.

```
$ CREATE/DIRECTORY (RET)
Device and UFD? [KIZZY] (RET)
```

This example creates directory [KIZZY] on the default device.

```
$ CREATE/DIRECTORY (RET)
Device and UFD? DY2: (RET)
```

This example creates a directory on the volume mounted on DY2: with the same group and member number as the default directory.

```
$ CREATE/DIRECTORY/PROTECTION:(S:RWED,O:R,G:R,W:R) (RET)
Device and UFD?DU0:[MOUSE] (RET)
```

This example creates directory [MOUSE] on the volume mounted on device DU0:. The protection code is the protection code applied to the actual file containing the directory, DU0:[000000]MOUSE.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 4.6.

## DELETE/DIRECTORY

### 4.7.3 DELETE/DIRECTORY

DELETE/DIRECTORY deletes a directory on a Files-11 volume and removes the directory from the volume's Master File Directory (MFD).

Privileged users can delete directories on volumes mounted on any device. Nonprivileged users can delete directories only on volumes mounted on their own private (allocated) devices.

#### Format

```
DELETE/DIRECTORY
Directory? [ddnn:] [directory]

DELETE/DIRECTORY [ddnn:][directory]
```

#### Parameters

[ddnn:] [directory]

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

ddnn:

Specifies the device on which the volume you want to delete the directory is mounted. This parameter defaults to SY0:.

The volume must be a mounted, Files-11 volume.

[directory]

Specifies the name of the directory you want to delete. This parameter must be specified.

The brackets are required syntax.

#### Examples

```
$ DELETE/DIRECTORY (RET)
Directory? DU1:[TSTSYS] (RET)
```

This example deletes directory [TSTSYS] on the volume mounted on device DU1:. Nonprivileged users can delete directories only on volumes mounted on a device they have allocated. Privileged users can delete directories on volumes mounted on any device.

```
$ DELETE/DIRECTORY DU1:[TSTSYS] (RET)
```

This example is identical to the previous one.

```
$ DELETE/DIRECTORY [EXTRA1] (RET)
```

This example deletes directory [EXTRA1] on the default device.

## EDIT

### 4.7.4 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

### 4.7.4.1 EDIT[/EDT]

EDIT invokes EDT, the DEC standard Editor. See the EDT Editor Manual for more information.

#### Format

```
EDIT[/EDT][qualifier[s]] infile
```

```
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 want 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 you inadvertently typed 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 want simply to read the file or to 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. And, if you

## EDIT/EDT (Cont.)

specified a name for the journal file other than 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 the 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 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.

If you want 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.

## EDIT/EDT (Cont.)

### 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

### 4.7.4.2 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.

#### Command 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 eight characters.

**EDIT/SLP (Cont.)**

The audit trail itself is defined from within SLP.

**/[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 want 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]**

The TRUNCATE qualifier 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 the /TRUNCATE qualifier without

## EDIT/SLP (Cont.)

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.

### Notes

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 RSX-11M/M-PLUS Utilities Manual for the alternate command format.

### 4.7.4.3 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/M-PLUS 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 want to access.

**EDIT/SLP (Cont.)****/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**

Specifies functions of the unsupported editor, Text Editor and Corrector (TECO).

**/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/M-PLUS 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 want 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 want 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.

## EDIT/SLP (Cont.)

### 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 (RET)
```

This example invokes SOS for a file that already exists.

```
$ EDIT/SOSLONNNY.MAC (RET)
Input: DB0:[7,7]LONNNY.MAC;1 (RET)
00100
```

This example invokes SOS for a file that is being created.

```
$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-PLUS systems. Commands to access these editors are included only for the convenience of users. No support is implied.

## 4.8 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****4.8.1 DIRECTORY**

DIRECTORY displays information on files in directories.

**Format**

DIRECTORY[/format-qual][/destination-qual] [filespec[s]]

**Format Qualifiers**

/FULL  
/BRIEF  
/FREE [ddnn:]  
/SUMMARY  
/ATTRIBUTES

**Destination Qualifiers**

/PRINTER  
/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; 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 more than one directory, 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 and detail of the directory listing.
2. Destination qualifiers control where the command output is sent.
3. Other qualifiers modify the list of files by creation date or exclusion, or have special uses.

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
- File organization

**DIRECTORY (Cont.)**

- 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-11 files. See the RMS-11 documentation supplied with your system.

**Destination Qualifiers**

These qualifiers direct output to an output file or to the line printer.

**/OUTPUT:filespec**

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

**/PRINTER**

Specifies that the output of the DIRECTORY command be printed on your system's line printer.

**Other Qualifiers**

**/DATE:dd-mmm-yy**

Specifies that you want the DIRECTORY command to list only files created on the date given.

**/SINCE:dd-mmm-yy**

Specifies that you want the DIRECTORY command to list only files created on or since the date given.

**/THROUGH:dd-mmm-yy**

Specifies that you want 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 want the DIRECTORY command to list only files created within that range.

## DIRECTORY (Cont.)

/TODAY

Specifies that you want the DIRECTORY command to list only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you want 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 with 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 

Directory DU0:[DOCDAD]  
1-JUN-85 14:16

A.A;1	1.		25-SEP-8412:29
ANIML.BAS;1	6.		24-MAR-8517:59
AZ.CMD;5	1.		02-APR-8513:03
BYE.CMD;1	1.		25-SEP-8412:29
CLEAN.CMD;1	1.		10-JAN-8508:29
DUN.CMD;4	1.		23-JAN-8508:05
EDT.CMD;22	1.		07-NOV-8415:56
LOX.CMD;1	1.		27-APR-8510:21
LOGIN.CMD;6	1.		06-APR-8515:56
TI.CMD;4	1.		30-APR-8508:36
.	.		.
.	.		.
F.TSK;1	4.	C	25-SEP-8412:30
G.TSK;1	4.	C	25-SEP-8412:30
PONG.TSK;2	12.	C	25-SEP-8412:31
BUZZ.TXT;2	4.		12-MAR-8509:13
IZZY.TXT;1	2.		06-MAR-8514:33
JIVE.TXT;1	1.		16-NOV-8413:23

Total of 2906./3043. blocks in 160. files

This example shows a directory listing in the standard format.

## DIRECTORY (Cont.)

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 three 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 4.9.4 for information on the UNLOCK command.

\$ DIRECTORY/SUMMARY (RET)

Storage used/allocated for Directory DU0:[DOCDAD]  
1-JUN-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)

DU0: has 12077. blocks free, 9523. blocks used out of 21600.  
Largest contiguous space = 10610. blocks  
3702. file headers are free, 298. headers used out of 4000.

This example displays information about the default device.

\$ DIRECTORY/FREE LBO: (RET)

DU0: has 12077. blocks free, 9523. blocks used out of 21600.  
Largest contiguous space = 10610. blocks  
3702. file headers are free, 298. headers used out of 4000.

This example displays information about the pseudo device LBO:. Notice that the display gives the physical device name for the pseudo device.

\$ DIR TI.CMD (RET)

Directory DU1:[DOCDAD]  
1-MAY-85 14:17

TI.CMD;4                      1.                      30-APR-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.

**DIRECTORY (Cont.)**

```
$ DIR TI.CMD;* (RET)
```

```
Directory DU1:[DOCDAD]
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

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 DU0:[DOCDAD]
1-MAY-85 14:17
```

TI.CMD;4	(300,56)	1./5.	30-APR-85
08:36			
[303,5]	[RWED,RWED,RWED,R]	03-MAY-85	09:58(2.)

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, and the second is the sequence number of the file. The UIC is the UIC of the owner, the protection status of the file, and the final column on the second row is the last revision date and revision level. The revision date is the date the file was last manipulated without a new version being created (for example, appending one file onto another). See the notes at the end of this section.

Protection status is listed in the order of system, owner, group, world. See Section 4.6 for an explanation of protection status and how it can be altered.

```
$ DIRECTORY/BRIEF *.CMD (RET)
```

```
Directory DU1:[DOCDAD]
```

```
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
TI.CMD;4
```

## DIRECTORY (Cont.)

This example displays brief information on the most recent version of all files of the type .CMD in the default directory.

\$ DIRECTORY/TODAY (RET)

Directory DU2:[DONNAX]  
5-JUN-85 13:44  
Day of 5-JUN-85

2051CH4.MEM;1	339.	05-JUN-85 10:11
NEWINSDC3.TXT;7	10.	05-JUN-85 10:05
2051CH4.MEM;2	320.	05-JUN-85 10:12
FUT.MAI;1	1.	05-JUN-85 13:05
NEWINSDC3.TXT;6	9.	05-JUN-85 09:55
NEWINSDC3.TXT;10	10.	05-JUN-85 10:39
NEWINSDC4.TXT;3	7.	05-JUN-85 11:46
FRK.CMD;6	1.	05-JUN-85 13:43
NEWINSDC4.TXT;4	11.	05-JUN-85 13:43
LOG.LOG;1	0.	05-JUN-85 13:43

Total of 708./736 blocks in 11. files

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 the next example.

\$ DIRECTORY/TODAY/EXCLUDE:\*.TXT;\* (RET)

Directory DU2:[OLDTIM]  
5-JUN-85 13:45  
\*.TXT;\* excluded  
Day of 5-JUN-85

2051CH4.MEM;1	339.	05-JUN-85 10:11
2051CH4.MEM;2	320.	05-JUN-85 10:12
FUT.MAI;1	1.	05-JUN-85 13:05
FRK.CMD;6	1.	05-JUN-85 13:43
LOG.LOG;1	2.	05-JUN-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 .TXT 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

## DIRECTORY (Cont.)

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, [ELBODGO]*.TXT (RET)
```

```
Directory DU1:[LAWRENCEP]
20-MAY-85 09:39
```

CLOCK.TSK;3	8.	C	15-APR-85	08:55
F.TSK;1	4.	C	25-MAR-85	12:30
G.TSK;1	4.	C	25-MAR-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 DU1:[ELBODGO]
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 [LAWRENCEP] requested directory information files in two directories. The default directory is applied to the first files named, the \*.TSK files; these are listed first in the display. The second set of files displayed, the \*.TXT files, are from directory [ELBODGO], which was explicitly specified in the command. Note the locked file. Compare this example with the next one.

```
$ DIR [ELBODGO]*.TXT, *.TSK (RET)
```

```
Directory DU1:[ELBODGO]
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	11-MAY-85	09:08
PONG.TSK;2	12.	C	13-MAY-85	14:13
YCLOCK.TSK;6	9.	C	08-APR-85	12:17
STAR.TSK;2	102.	C	13-APR-85	09:07
TICTAC.TSK;5	19.	C	01-APR-85	11:31
VTCHS.TSK;12	26.	C	06-MAY-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
```



**DIRECTORY (Cont.)**

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 [LAWRENCEP], issued the command but this time the directory [ELBODGO] is specified first, for the \*.TXT files. Even though no directory was specified for the \*.TSK files, these files are also listed from directory [ELBODGO]---not from [LAWRENCEP]. 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.

**Notes**

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 revised by the system or a task, such as for editing. The decimal number in parentheses following the second date is the number of times the file has been revised 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 [000000]. All directories on the volume appear in this directory in the same form. The directory for directory [FROGMAN] is called FROGMAN.DIR; the directory for directory [007040] 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. 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.

## DELETE

### 4.8.2 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

```
/LOG
/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 want 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.

/QUERY

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 six characters:

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

**DELETE (Cont.)**

- N - Does not delete file named and goes on to next file.
- 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 more files are deleted.
- CTRL/Z - Same effect as Q (Quit).
- RETURN - Same effect as N (No).

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

/DATE:dd-mmm-yy

Specifies that you want the DELETE command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you want the DELETE command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you want 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 that range.

/TODAY

Specifies that you want the DELETE command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you want the DELETE command not 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 with a wildcard (\*).

/NOWARNINGS

Suppresses error messages resulting from the command.

## DELETE (Cont.)

### Examples

```
$ DELETE (RET)
File(s)? *.DAT; (RET)
```

This example deletes all versions of all files of the type .DAT.

```
$ 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 file name or type is specified for version 3, and no name for .TXT;5. This form requires user confidence.

```
$ DELETE/QUERY (RET)
File(s)? *.TMP;* (RET)
Delete file DU1:[KATHYED]OGRE.TMP [Y/N/G/Q]? Y (RET)
Delete file DU1:[KATHYED]TROLL.TMP;1 [Y/N/G/Q]? Y (RET)
Delete file DU1:[KATHYED]ORC.TMP;1 [Y/N/G/Q]? Y (RET)
Delete file DU1:[KATHYED]ELF.TMP;1 [Y/N/G/Q]? N (RET)
Delete file DU1:[KATHYED]HOBBIT.TMP;1 [Y/N/G/Q]? N (RET)
Delete file DU1:[KATHYED]SNIPE.TMP;1 [Y/N/G/Q]? G (RET)
```

The following files have been deleted:

```
DU1:[KATHYED]SNIPE.TMP;1
DU1:[KATHYED]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 *.DOC (RET)
Delete file DU0:[ANNEDIT]WITCH.DOC;2 [Y/N/G/Q]? Y (RET)
Delete file DU0:[ANNEDIT]DRY.DOC;4 [Y/N/G/Q]? N (RET)
Delete file DU0:[ANNEDIT]PAYCHECK.DOC;1 [Y/N/G/Q]? Y (RET)
Delete file DU0:[ANNEDIT]LOADING.DOC;3 [Y/N/G/Q]? Y (RET)
Delete file DU0:[ANNEDIT]LOADING.DOC;4 [Y/N/G/Q]? Y (RET)
Delete file DU0:[ANNEDIT]LOADING.DOC;5 [Y/N/G/Q]? Y (RET)
Delete file DU0:[ANNEDIT]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:

```
DU1:[JIMPAR]RANGER.LST;1
DU1:[JIMPAR]TONTTO.LST;1
DU1:[JIMPAR]REID.LST;1
DU1:[JIMPAR]SILVER.LST;1
DU1:[JIMPAR]SCOUT.LST;1
DU1:[JIMPAR]HORNET.LST;1
DU1:[JIMPAR]KATO.LST;1
```

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.

## DELETE (Cont.)

### Notes

You must have DELETE access to delete a file.

If you want 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-PLUS systems, you have DELETE access to your own files and all files in directories whose owners have the same group number, the first of the two numbers in the UIC.

## PURGE

### 4.8.3 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 or files to be purged.

Because of the nature of the command, you cannot include a version number in the file specification. Multiple file specifications must be separated by commas. Wildcards can be substituted for directory, 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 the second example.

**PURGE (Cont.)**

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

/DATE:dd-mmm-yy

Specifies that you want the PURGE command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you want the PURGE command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you want 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 want the PURGE command to affect only files created within that range.

/TODAY

Specifies that you want 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 not 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 with 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.

**PURGE (Cont.)**

\$ **DIRECTORY ASDIC.TM1:\*, ELPASO.TEX;\*** (RET)

Directory DU1:[BRUCEW]  
20-JUN-85 13:44

ASDIC.TM1;1	1.	20-JUN-85 13:41
ASDIC.TM1;2	1.	20-JUN-85 13:41
ASDIC.TM1;3	1.	20-JUN-85 13:41
ASDIC.TM1;4	0.	20-JUN-85 13:41
ASDIC.TM1;5	1.	20-JUN-85 13:42
ELPASO.TEX;1	1.	20-JUN-85 13:42
ELPASO.TEX;2	1.	20-JUN-85 13:42
ELPASO.TEX;3	1.	20-JUN-85 13:42
ELPASO.TEX;5	1.	20-JUN-85 13:43
ELPASO.TEX;6	1.	20-JUN-85 13:43

Total of 9./45. blocks in 10. files

\$ **PURGE/LOG/KEEP:3 ASDIC.TM1** (RET)

The following files have been deleted:

DUO:[GLENN]ASDIC.TM1:1  
DUO:[GLENN]ASDIC.TM1:2

\$ **PURGE/LOG/KEEP:3 ELPASO.TEX** (RET)

The following files have been deleted:

DUO:[GLENN]ELPASO.TEX;1  
DUO:[GLENN]ELPASO.TEX;2  
DUO:[GLENN]ELPASO.TEX;3

\$ **DIRECTORY ASDIC.TM1:\*, ELPASO.TEX;\*** (RET)

Directory DU0:[GLENN]  
20-JUN-85 13:46

ASDIC.TM1;3	1.	20-JUN-85 13:41
ASDIC.TM1;4	0.	20-JUN-85 13:41
ASDIC.TM1;5	1.	20-JUN-85 13:42
ELPASO.TEX;5	1.	20-JUN-85 13:43
ELPASO.TEX;6	1.	20-JUN-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****4.8.4 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
/REPLACE
/NONEW_VERSION
/NOWARNINGS
/OVERLAY
/PRESERVE_DATE
/REWIND
/SHARED
/OWN
/BLOCK_SIZE: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 and an explicit output file, they will be concatenated to the output file in the order that you specify them.

**outfile**

Specifies a single output file to which the input file or files are copied.

## COPY (Cont.)

You must have WRITE access to the directory to which you want 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.

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 unless you use the /REPLACE qualifier, which causes the input file to replace the existing file.

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.

#### /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.

#### /NONEW\_VERSION

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

**COPY (Cont.)****/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 before is destroyed, but the file ID is unchanged. The output file must exist before you issue the command.

**/PRESERVE\_DATE**

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

**/REWIND**

Causes the tape to rewind before starting the operation. This qualifier is for magnetic tapes only.

**/SHARED**

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

**/OWN**

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

**/BLOCK\_SIZE:n**

Defines block size for magnetic tape. The argument n can be octal or decimal; if decimal, you terminate n with a decimal point. For example:

**/BLOCK\_SIZE:25.**

**/DATE:dd-mmm-yy**

Specifies that you want the COPY command to affect only files created on the date given.

**/SINCE:dd-mmm-yy**

Specifies that you want the COPY command to affect only files created on or since the date given.

**/THROUGH:dd-mmm-yy**

Specifies that you want the COPY command to affect only files created on or before the date given.

## COPY (Cont.)

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

Specifies that you want the COPY command to affect only files created within that range.

/TODAY

Specifies that you want the COPY command to affect only files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you want the COPY command not 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 with a wildcard (\*).

### Examples

```
$ COPY (RET)
From? [KILIKI]TSKBLD.CMD (RET)
To? TSKBLD.CMD (RET)
```

This example copies TSKBLD.CMD from [KILIKI] to the current directory and device. The file is still owned by the source UIC.

```
$ COPY *.BAS DU1: (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 DU1:.

```
$ COPY/OWN TSKBLD.CMD [MONGO]BLDFIL.CMD (RET)
```

This example copies TSKBLD.CMD from the current directory and device to [MONGO], assigns ownership to the destination UIC, and also changes the file's 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 (Cont.)

\$ COPY CHARLA.DMP TT4: (RET)

This example prints a copy of CHARLA.DMP on TT4:. 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.

**Notes**

C is the short form of copy.

COPY does not affect file organization. If you want to change file organization, use the CONVERT command, Section 4.8.6.

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

### 4.8.5 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
/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 the 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 want the RENAME command to affect only files created on the date given.

**RENAME (Cont.)**

`/SINCE:dd-mmm-yy`

Specifies that you want the RENAME command to affect only files created on or since the date given.

`/THROUGH:dd-mmm-yy`

Specifies that you want 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 want the RENAME command to affect only files created within that range.

`/TODAY`

Specifies that you want the RENAME command to affect only files created on the same day the command is being issued.

`/EXCLUDE:filespec`

Specifies that you want the RENAME command not 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 with a wildcard (\*).

`/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 MAIN.TSK;* SUB.* (RET)
```

## RENAME (Cont.)

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.* 
```

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****4.8.6 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 are 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-11 documentation supplied with your system for more information.

No wildcards are permitted.

## CONVERT (Cont.)

### Command Qualifiers

#### /[NO]APPEND

Specifies that you want 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. The argument 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

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 is the same size, RMSCNV performs a straightforward copy. 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.

This qualifier is the equivalent of the /ID switch in an RMSCNV command line.

The default is /NOIDENTIFICATION.

## CONVERT (Cont.)

### /INDEXED

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.

### /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. The argument 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:

- The command string, in RMSCNV format. This will not be the same as DCL format.
- Copies of all error messages produced during execution.
- 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.

### /[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.

**CONVERT (Cont.)**

This is the equivalent of the /MA switch in an RMSCNV command line.

The default for nonsequential file organization is /NOMASS\_INSERT.

**/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. Use this qualifier only when the output file specifies fixed-length records. The default is /NOPAD, meaning record length remains the same.

You can pad with any single character as follows:

- If you specify /PAD without an argument, the pad character is null.
- If you specify /PAD:R, then the pad character is R. If you specify /PAD:r, the pad character is still R, not r. This rule applies for all the letters of the alphabet, the numbers 0 through 9, and any other printing character except the number sign (#) or the quotation mark ("), which cannot be used.
- If you specify /PAD:"R", then the pad character is R. If you specify /PAD:"r", then the pad character is r. The quotation marks permit you to pad with lowercase letters, the #, or a space.
- If you specify /PAD:#000, the pad character is null, because 000 is the octal value of the null character. In fact, the number can be any octal number from 000 through 377, representing all of the ASCII character set, plus the octal numbers 200 through 377. With the # in place, you can use the # as a pad character by specifying 043 and the quotation mark (") by specifying 042.

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 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.

**CONVERT (Cont.)**

RMSCNV reads records from the input file and writes them into successive record cells of the output file, beginning with cell one. 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.

**/[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.

## CONVERT (Cont.)

/[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.

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)
```

## CONVERT (Cont.)

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/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 more information.

RMS and FCS sequential files are compatible. Sequential files created by FCS can be input files for CONVERT, and sequential files created by CONVERT can be read by FCS.

### 4.9 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	Alters the contents of certain files
DIFFERENCE	Displays the differences between two files
UNLOCK	Unlocks locked files

## TYPE

### 4.9.1 TYPE

TYPE displays 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 displayed on your terminal.

You can specify any file to which you have READ access.

The file name and type must be specified explicitly or with a wildcard (\*). If no version is specified, the most recent version is displayed. A wildcard in any field displays 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 want the TYPE command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you want the TYPE command to affect only files created on or since the date given.



**TYPE (Cont.)**

**/THROUGH:dd-mmm-yy**

Specifies that you want the TYPE 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 TYPE command to affect only files created within that range.

**/TODAY**

Specifies that you want the TYPE command to affect only files created on the same day the command is being issued.

**/EXCLUDE:filespec**

Specifies that you want the TYPE command not 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 with 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/INQUIRE
ASSIGN DB1: SY0:
@AZ
```

This example prints the file LOGIN.CMD from the current device and directory on your terminal.

```
$ TYP [DALE]LOGIN.CMD (RET)
TYPE LOGIN.TXT
SET TERMINAL/INQUIRE
```

This example prints the file LOGIN.CMD from the current device and directory [DALE] on your terminal.

### TYPE (Cont.)

\$ TYPE [\*] \*.FIL;\* RET

This is the beginning of a file named ARTHUR.FIL.  
It goes on for several lines.

•

THIS MARKS THE INITIATION OF A FILE CALLED ALBERT.FIL  
IT TOO CONTINUES AT SOME LENGTH.

•

Now comes a file that starts like this and is called Z.FIL.  
And some more.

•

At last, we reach the beginning of a file named FILLIE.FIL.  
Again, some more text and then

•

The last two lines of FILLIE.FIL.  
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

This example prints on your terminal all versions of all files having the type `.FIL` located on the current device.

## Notes

In general, RSX-11M-PLUS documentation uses the term "print" to refer to any output sent to a terminal and "type" to refer to any user input from a terminal. The name of the TYPE command is an exception to this rule.

If you are typing multiple files, the CTRL/O feature works in a slightly different fashion. When you enter a CTRL/O, output is discarded until you type another **CTRL/O** or until the beginning of the next file is reached, whereupon output is sent to your terminal again. This leaves you free to use **CTRL/O** on a file-by-file basis.

You can display any file using TYPE, but normally only ASCII text files are readable.

**APPEND****4.9.2 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 want the APPEND command to affect only files created on the date given.

**APPEND (Cont.)**

`/SINCE:dd-mmm-yy`

Specifies that you want the APPEND command to affect only files created on or since the date given.

`/THROUGH:dd-mmm-yy`

Specifies that you want 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 want the APPEND command to affect only files created within that range.

`/TODAY`

Specifies that you want the APPEND command to affect only files created on the same day the command is being used.

`/EXCLUDE:filespec`

Specifies that you want the APPEND command not 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 with 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.

`/SHARED`

Specifies that you want others to be able to access the file while your append operation is underway.

**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)
```

This example 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.

## APPEND (Cont.)

### 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

### 4.9.3 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

## DIFFERENCES (Cont.)

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)
```

```
/IGNORE:(arg[,s])
  BLANK_LINES
  COMMENTS
  FORM_FEEDS
  SPACING
  TRAILING_BLANKS
```

Specifies that you want 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.

## DIFFERENCES (Cont.)

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.

/[NO]NUMBERS

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 a 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>
        EXST$C EX$SUC          ; PRINT THE STRING
        EXST$C EX$SUC          ; SUCCESSFUL EXIT IN ALL CASES

.END    START                  ; SETUP THE STARTING ADDRESS
```



## DIFFERENCES (Cont.)

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/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$$
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\$\$ 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$$
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.

```
$ 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
```

## DIFFERENCES (Cont.)

\*\*\*\*\*

```

1) DB2:[303,5]TOM.MAC;1
  11          QIOW$C          IO.WVB,1,1,,, <STRING,LENGTH,40>
*****
2) DB2:[303,5]TOM.MAC;3
  12
  13          QIOW$C          IO.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

### 4.9.4 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 want to unlock.

#### Command Qualifiers

/DATE:dd-mmm-yy

Specifies that you want the UNLOCK command to affect only files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you want the UNLOCK command to affect only files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you want 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 want the UNLOCK command to affect only files created within that range.

/TODAY

Specifies that you want the UNLOCK command to affect only files created on the same day the command is being issued.

## UNLOCK (Cont.)

/EXCLUDE:filespec

Specifies that you want the UNLOCK command not 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 with a wildcard (\*).

/NOWARNINGS

Suppresses error messages resulting from the command.

**Example**

```
$ Directory TOPEKA.KAN (RET)
Directory DU0:[PBEZ]
5-JUL-85 13:13

ELIOT.MAN;3      32.      L      25-JUN-85 12:29

Total of 32./40. blocks in 1. files

$ UNLOCK TOPEKA.KAN (RET)

$ DIRECTORY TOPEKA.KAN (RET)
Directory DU0:[PBEZ]
5-JUL-85 13:14

TOPEKA.KAN;3      32.      25-JUN-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.

**4.10 PRINT AND QUEUE MANAGER COMMANDS**

The following section describes the PRINT command as well as all the nonprivileged Queue Manager commands.

# PRINT

## 4.10.1 PRINT

PRINT queues files for printing on a line printer. It can also be used to queue jobs for other output devices that are serviced by the Queue Manager. 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
/[NO]TRANSFER
/DEVICE:ddnn:
/[NO]HOLD
/JOB COUNT:n
/[NO]JOB PAGE
/QUEUE:queuename
/[NO]UPPERCASE
/[NO]LOWERCASE
/[NO]WIDE
/PAGE COUNT:n
/NAME:jobname
/PRIORITY:n
/FORMS:n
/LENGTH:n
/[NO]RESTART
/[NO]FLAG PAGE
/AFTER:(dd-mmm-yy hh:mm)
          (mm/dd/yy hh:mm)
/AFTER:TOMORROW
/[NO]WIDE
```

#### File Qualifiers

```
/[NO]DELETE
/COPIES:n
/[NO]TRANSFER
```

### Parameter

filespec[s]

Specifies the file or files to be printed on the line printer.

Multiple file specifications must be separated by commas.

A 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 wildcard (%) accepted.

If your file specification includes no file type, the default file type is .LST.

**PRINT (Cont.)****Command Qualifiers****/[NO]DELETE**

Specifies that files be deleted or not deleted from the directory after they are printed. The default is /NODELETE.

The /[NO]DELETE qualifier to the file specification 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 want 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.

**/[NO]TRANSFER**

Transfers spooling of a file from the original device, even if that device is not public. Normally, files on nonpublic devices are copied to pseudo device SP: before spooling.

This switch can be overridden on the file level. The default is /NOTRANSFER.

See the examples.

**/DEVICE:ddnn:**

Specifies that you want the file printed on a particular device.

**/[NO]HOLD**

Specifies that you want the QMG print job to be held in its queue until you release it. See the RELEASE/JOB or RELEASE/ENTRY commands in the next section. The default is /NOHOLD.

**/JOB\_COUNT:n**

Specifies the number of sets of files you want 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 of the files named in order, followed by another set, and so on. If you want to print multiple copies of one or more files, but not in sets, use the /COPIES qualifier. See the examples.

## PRINT (Cont.)

[illegible]

**Figure 4-2: Job Flag Page**



**PRINT (Cont.)**

01234567890123456789  
01234567890123456789  
01234567890123456789

14:01:58  
14:01:58  
14:01:58

4-MAR-85  
4-MAR-85  
4-MAR-85

DRS:(301,7)OISKS.CMD;1  
COPY 1 OF 1  
DELETION NOT SPECIFIED

RSX-11M-PLUS V3.0  
RSX-11M-PLUS V3.0  
RSX-11M-PLUS V3.0

0123456789 123456789  
0123456789 123456789  
0123456789 123456789

[illegible][illegible]

01234567890123456789  
01234567890123456789  
01234567890123456789

14:01:58  
14:01:58  
14:01:58

4-MAR-85  
4-MAR-85  
4-MAR-85

DR5:[301,71DISKS.CMD;1  
COPY 1 OF 1  
DELETION NOT SPECIFIED

RSX-11M-PLUS V3.0  
RSX-11M-PLUS V3.0  
RSX-11M-PLUS V3.0

001234567890123456789  
001234567890123456789  
001234567890123456789

**Figure 4-3: File Flag Page**

**PRINT (Cont.)****/[NO]JOB\_PAGE**

Specifies whether you want 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 want to include a job flag page. See the /[NO]FLAG\_PAGE qualifier for controlling file flag pages. Figures 4-2 and 4-3 show the difference between job flag pages and file flag pages.

**/QUEUE:queue name**

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.

**/[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:job name**

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, the value for 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 longer is printed first.

**PRINT (Cont.)****/FORMS:n**

Specifies the type of form on which the job must be printed. The argument *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**.

**/[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.

The **/NORESTART** qualifier 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. Figures 4-2 and 4-3 show the difference between the job flag page and the file flag page.

**PRINT (Cont.)**

```
/AFTER:arg
      (dd-mmm-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-mmm-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. The parameter yy must be between 77 and 92. If you leave it out, the year defaults to the current year.

The parameter hh:mm:ss is the hour, minute, and second. The parameter ss need not be given. It defaults to 00. In fact, field ss is not actually used. If you do specify ss, it is checked for syntax only. The Queue Manager always takes a few seconds to start the job. If you leave out the clock field, the time defaults to 00:00:00 on the given date.

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, and if you have to print some output from that file, you can include PRINT/AFTER:TOMORROW in the command file. This qualifier gets around the problem that it is always after midnight.

```
/[NO]WIDE
```

Specifies that the file must be printed on a processor that has been initialized /WIDE. The default is /NOWIDE.

**File Qualifiers**

```
/COPIES:n
```

Overrides the /COPIES command qualifier when attached to a particular file specification.

## PRINT (Cont.)

/[NO]DELETE

Overrides the /[NO]DELETE command qualifier when attached to a particular file specification.

/[NO]TRANSFER

Overrides the /[NO]TRANSFER 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 the next section.

```
$ PRINT/COPIES:20 (RET)
File(s)? PASHA.CMD (RET)
PRI -- Job 706, name "PASHA", submitted to queue "PRINT "
```

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 want 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, 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 (Cont.)

```
$ 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.

```
$ PRI/COPIES:20 MOB.COR, RIFF.COR, RAB.COR/COP:19, PROXOR (RET)
PRI -- Job 77, name "MOB ", submitted to queue "PRINT "
```

This example prints 20 copies each of MOB.COR, and RIFF.COR, 19 copies of RAB.COR, and then 20 copies of PROXOR.

```
$ 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 directory.

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-84 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 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.

PRINT is to print jobs as SUBMIT is to batch jobs. The PRINT command defines QMG print jobs.

**PRINT (Cont.)**

Once your job is in a queue, you can modify some job attributes with SET QUEUE. See the next section.

The Queue Manager automatically deletes .PMT and .DMP files after they are printed unless specifically told not to (by using the /NODELETE qualifier).

For more information on the PRINT command, see the RSX-11M/M-PLUS Batch and Queue Operations Manual.

File qualifiers are used to override the corresponding command qualifiers when you have given multiple file specifications as parameters to the PRINT command.

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 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.

If you use the /NOTTRANSFER qualifier, you must leave the volume mounted until the print job is finished. Under the default, /NOTTRANSFER, your files are copied to another location and the volume can be dismounted. If you do not want your files to be accessible to other system users, use the /NOTTRANSFER qualifier. This qualifier is not recommended when the original device is slow, such as diskettes. On the other hand, if there is no space on the pseudo device SP: (usually the system disk), you can use the /NOTTRANSFER qualifier rather than waiting for space on SP:.

The flag pages and the job pages have different formats for the banner text. The job 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 the next section for information on how to use this number in other commands.

#### **4.10.2 Queue Manager Commands for the User**

This section contains Queue Manager commands that are generally intended for the average user.

## DELETE

### 4.10.2.1 DELETE Command

The DELETE command is used to delete jobs from a queue or files from a job. The command allows the deletion of a job in queue by specifying either the job's entry number or job name. You can also delete a single file in a job by specifying the /FILE\_POSITION:n qualifier.

#### Format

```
DELETE/ENTRY:nnn[/FILE_POSITION:n]
```

```
DELETE/JOB queueName jobName[/FILE_POSITION:n]
```

#### Parameters

##### queueName

Specifies the name of the queue containing the job you want to delete. Used with the /JOB qualifier only.

##### jobName

Identifies a job to be deleted from the specified queue. Used with the /JOB qualifier only.

The same job name may appear in a queue more than once. HELD or ACTIVE jobs will be deleted before WAITING jobs.

The job name appears on the first line of the SHOW QUEUE entry and consists of the UIC plus the name of the first file in the job, or a name specified with the /NAME qualifier to PRINT or SUBMIT.

Nonprivileged users need never specify the UIC. Privileged users may need to specify the UIC if issuing the command under a UIC different from that used when the job was entered.

#### Command Qualifiers

##### /ENTRY:nnn

Deletes a queue entry by number. The number is unique. The entry number replaces both the queue name and the job name.

##### /FILE\_POSITION:n

Identifies the file by the file's position in the job.



**DELETE (Cont.)****Examples**

\$ **DELETE/ENTRY:301** (RET)

This command deletes the job from the queue by referencing the job's unique entry number (in this example, job number 301). Jobs may share the same job name but never the same job entry number. When you delete a job by its entry number, you are deleting that particular job with greater confidence. It is possible to delete the wrong job when a job is referenced by its job name. Deleting a job by its unique entry number protects you from such mistakes.

\$ **DELETE/JOB LP0 ANGEL** (RET)

This command deletes the job ANGEL by first referencing the job's queue, LP0, and then deleting the job ANGEL from LP0.

\$ **DELETE/ENTRY:301/FILE\_POSITION:2** (RET)

This example deletes only the second file appearing in job 301. You may only refer to the file you want to delete by the file's position in the print job (the numbered order in which you entered the file specification). You cannot specify the file you want to delete by referring to its file specification.

**Notes**

Associating a particular file specification with the numbered order in which it was entered into a queue can be accomplished by entering a SHOW QUEUE command, and examining the output on your terminal.

Using the /ENTRY:nnn qualifier to delete a job saves keystrokes. You are also deleting a job with greater confidence. Remember, a job may share a job name but never a job entry number. If, for example, you decided to delete a job by referring to the job name, and you have several jobs sharing the job name, you would succeed in deleting only the first occurrence of the job and not necessarily the job that you wanted to delete.

DELETE/QUEUE works on all categories of jobs in any queue. You can delete ACTIVE jobs, WAITING jobs, TIME-BLOCKED jobs, or HELD jobs.

There can be more than one job with the same name from the same UIC. The DELETE/QUEUE command will delete the first job of a given name in the queue.

## SHOW QUEUE

### 4.10.2.2 SHOW QUEUE

SHOW QUEUE displays information about QMG print jobs and RSX-11M-PLUS batch jobs.

#### Format

```
SHOW QUEUE [queueName][qualifier[s]]
```

#### Command Qualifiers

```
/BATCH
/BRIEF
/DEVICE
/ENTRY:nnn
/FILES
/FORMS[:n]
/FULL
/NAME:jobname
/OWNER_UIC:uic
/PRINT
```

#### Parameter

##### queueName

Displays QMG information for a single queue, either print or batch. The main print queue is called PRINT. Other print queues have names determined by the system manager.

#### Command Qualifiers

##### /FULL

Displays detailed information about queues, queue assignments, jobs, the attributes of jobs in queues, and files that compose jobs in queues.

##### /FILES

Displays information about queues, queue assignments, jobs in queues, and files that compose jobs in queues. The attributes of the jobs are not displayed. This display format is the default of SHOW QUEUE.

##### /BRIEF

Displays only queues, queue assignments, and jobs in queues.

You may examine specific attributes of queues by entering one of the following qualifiers.

##### /DEVICE

Limits information to print queues.

## SHOW QUEUE (Cont.)

/ENTRY:nnn

Limits information to a particular job entry referenced by the job's unique entry number.

/FORMS[:n]

Limits information to jobs that are to be printed on a specified form. If n is omitted, the display shows all jobs that are other than Form 0.

/NAME:jobname

Limits information to jobs with the specified jobname.

/OWNER UIC:uic

Limits information to jobs owned by the specified UIC.

/PRINT

Limits information to print queues. This qualifier produces the same information as the /DEVICE qualifier.

/BATCH

Limits information to batch queues.

## Examples

The following examples show the three different output displays (FULL, FILE, BRIEF):

```
$SHOW QUEUE/FULL (RET)

** PRINT QUEUES **
PRINT => LP0 (LP1)
  [7,25] ANGEL      ENTRY:22      FORM:0      ACTIVE ON LP0
        PRI:50     LENTO  PAGE:0    NORESTART  FLAG:JOB   NOLOWER  COP:1
        > 1 DU0:[GREGG]ANGEL.LST;1      COP:1      NODELETE
  [7,40] ANITA      ENTRY:24      FORM:0
        PRI:50     LEN:0   PAGE:0    NORESTART  FLAG:JOB   NOLOWER  COP:1
        1 DU0:[LAKE]ANITA.MEM;120      COP:1      NODELETE
LP0      => LP0
LP1      => (LP1)
** BATCH QUEUES **
BATCH    => BAP0
  [7,40] MJRA       ENTRY:23              ACTIVE ON BAP0
        PRI:50     NORESTART LOG PRINT:PRINT
        > 1 DU0:[7,40]MJRA.BAT;1              NODELETE
```

This display shows all information about the queues and jobs in queues. In this example, the Queue PRINT can send jobs to LP0 or LP1. LP1 has been stopped by the System Manager and is therefore not eligible to accept jobs. ( This is indicated by the processor name enclosed in parentheses.)

## SHOW QUEUE (Cont.)

There are two files to be printed in the Queue PRINT:

- [7,25]ANGEL.LST, entry number 22, which is currently being printed on LP0
- [7,40]ANITA.MEM, entry number 24, which is waiting to be printed

The batch job [7,40] MJRA, entry number 23, is running on batch processor BAP0.

The attributes of the job are indicated in the display, and the files that make up each job are also listed. The angle bracket (>) indicates which file of a job is currently being processed.

```

$ SHOW QUEUE (RET)

** PRINT QUEUES **
PRINT => LP0 (LP1)
  [7,25] ANGEL      ENTRY:22                ACTIVE ON LP0
      > 1 DU0:[GREGG]ANGEL.LST;1
  [7,40] ANITA      ENTRY:24
      1 DU0:[LAKE]ANITA.MEM;120
LP0      => LP0
LP1      => (LP1)
** BATCH QUEUES **
BATCH    |
  [7,40] MJRA      ENTRY:23                ACTIVE ON BAP0
      > 1 DU0:[7,40]MJRA.BAT;1

```

This display, the default format, shows the queues in the same state. It does not show the attributes of the job. It only shows the form number of the job and the attributes of the files if other than the default values were specified.

```

$ SHOW QUEUE/BRIEF (RET)

** PRINT QUEUES **
PRINT => LP0 (LP1)
  [7,25] ANGEL      ENTRY:22                ACTIVE ON LP0
  [7,40] ANITA      ENTRY:24
LP0      => LP0
LP1      => (LP1)
** BATCH QUEUES **
BATCH => BAP0
  [7,40]MJRA      ENTRY:23                ACTIVE ON BAP0

```

This display only shows the names, entry numbers, and status of the jobs in the queues. Also, if a form other than Form 0 is specified for a print job, the form will appear in the display.

## SHOW PROCESSOR

### 4.10.2.3 SHOW PROCESSOR

SHOW PROCESSOR displays information about the initialized characteristics of spooled devices and batch processors, printers, and other output devices under the control of the Queue Manager.

#### Format

```
SHOW PROCESSOR [processorname]
```

Displays information about all processors. This includes batch processors and spooled input devices.

```
SHOW PROCESSOR/BATCH
```

Shows all batch processors.

```
SHOW PROCESSOR/PRINT or SHOW PROCESSOR/DEVICE
```

Shows all print processors.

#### Example

```
$ SHOW PROCESSOR 
```

Information about print processors, batch processors, and spooled input devices is displayed at your terminal:

```
** SPOOLED DEVICES **
LP0      <= PRINT LP0
          LOWER          FORM:0    FLAG:2
          CURRENT JOB:    [CRAZED]FUP          ENTRY:59
LP1      (STOPPED) <= LP1
          NOLOWER        FORM:1    FLAG:1
** BATCH PROCESSORS **
BAP0     <= BATCH
```

This example shows all information about spooled devices and batch processors. In this example, spooled device LP0 can receive jobs from queues PRINT and LP0.

Spooled device LP1 has been stopped by the System Manager and is therefore not eligible for receiving jobs for printing.

The batch processor, BAP0, can only receive jobs from the Queue BATCH. There is no other batch processor.

LP0 is currently printing job [CRAZED]FUP, entry number 59.

## SET QUEUE

### 4.10.2.4 SET QUEUE

SET QUEUE modifies attributes given to print jobs, batch jobs, or files that compose jobs in queues. Such jobs and files have been entered in queues by either the PRINT command or the SUBMIT command. You cannot change the attributes of an active job.

#### Job Formats

```
SET QUEUE queueName jobName/qualifier[/qualifier[s]]
```

```
SET QUEUE/ENTRY:nnn/qualifier[/qualifier[s]]
```

#### Job Qualifiers

```
/AFTER:(dd-mmm-yy hh:mm)
```

```
/FORMS:n
```

```
/JOBCOUNT:n
```

```
/LENGTH:n
```

```
/LOWERCASE
```

```
/PAGE COUNT:n
```

```
/PRIORITY:n
```

```
/RELEASE
```

```
/[NO]RESTART
```

```
/UPPERCASE
```

#### File Formats

```
SET QUEUE /queueName jobName/FILE_POSITION:n/qualifier[/qualifier[s]]
```

```
SET QUEUE/ENTRY:nnn/FILE_POSITION:n/qualifier[/qualifier[s]]
```

#### File Qualifiers

```
/COPIES:n
```

```
/[NO]DELETE
```

#### Job Qualifiers

```
/AFTER:(dd-mmm-yy hh:mm)
```

Changes the time after which your job will be printed. The job will be BLOCKED until the time and date you specify. The job will not necessarily be printed at exactly the time you state, but will be eligible after the time you state.

If you do not supply the calendar field, the default is the current date. If you do not supply the clock field, the default is midnight on the date given in the calendar field.

**SET QUEUE (Cont.)**

If you supply both the clock and calendar fields, you must separate them with a space.

**/FORMS:n**

Changes the FORMS attribute of your print job. See Section 4.10.1 for an explanation of the FORMS attribute as it applies to print jobs.

**/JOBCOUNT:n**

Changes the number of copies of a job you want to have printed. See File Qualifiers for an explanation of how to modify the number of copies of a file contained inside a job. See Section 4.10.1 for an explanation of the COPIES attribute as it applies to print jobs.

**/LENGTH:n**

Changes the length of a logical page (number of lines per printed page) of your print job. A line printer will move to the top of a physical page every time n lines have been printed on a page. See Section 4.10.1 for an explanation of the LENGTH attribute as it applies to print jobs.

**/LOWERCASE**

Prints your job on a line printer initialized /LOWERCASE by your system manager.

**/PAGE\_COUNT**

Changes the limit on the number of pages your job may produce. See Section 4.10.1 for an explanation of the PAGE\_COUNT attribute as it applies to print jobs.

**/PRIORITY:n**

Changes the queue priority of a print job. Nonprivileged users may set priorities up through 150. Privileged users may set priorities up through 250. See Section 4.10.1 for an explanation of the PRIORITY attribute as it applies to print jobs.

**/RELEASE**

Unblocks a job that has been held in queue. SET QUEUE/RELEASE is a synonym for the RELEASE/JOB and RELEASE/ENTRY commands.

**/[NO]RESTART**

Changes the restartability of your job. If you specify /RESTART, your job will start again from the beginning if it is interrupted while ACTIVE. If you specify /NORESTART, the default, your job will pick up where it left off, if it is interrupted while ACTIVE.

**/UPPERCASE**

Changes the attribute of the job from lowercase to uppercase.

## SET QUEUE (Cont.)

### File Qualifiers

/COPIES:n

Changes the number of copies of a file you want to have printed within a single printing of a print job.

/[NO]DELETE

Changes the delete status of a single file contained in a print job.

/FILE POSITION:n

Changes the operation of /COPIES:n or /[NO]DELETE as they apply to a file contained in a job in queue. The number n refers to the file position in the job. Use SHOW QUEUE to determine its position.

### Example

```
$ PRINT/JOB COUNT:2 JILL.TXT,JOHN.TXT/COPIES:3 (RET)
```

The SHOW QUEUE/FULL command output might look like this:

```
** PRINT QUEUES **
PRINT => (LP0)
  [303,5] JILL ENTRY:696 FORM:0
        PRI:50 LEN:0 PAGE:0 NORESTART FLAG:JOB NOLOWER COP:2
        1 DUO:[DICK]JILL.TXT;1 COP:1
NODELETE
        2 DUO:[DICK]JOHN.TXT;1 COP: 3 NODELETE
LP0    => LP0
LP1    => LP1
```

To print two copies of the file JOHN.TXT instead of the three copies specified in the initial print command, you would enter the following SET QUEUE command:

```
$ SET QUEUE/ENTRY:696/FILE_POSITION:2/CO:2 (RET)
```

The SHOW QUEUE/FULL command output on your terminal would look like this:

```
** PRINT QUEUES **
PRINT => (LP0)
  [ 303,5] JILL ENTRY:696 FORM:0
        PRI:50 LEN:0 PAGE:0 NORESTART FLAG:JOB NOLOWER COP:2
        1 DUO:[DICK]JILL.TXT;1 COP:1 NODELETE
        2 DUO:[DICK]JOHN.TXT;1 COP : 2 NODELETE
LP0    => LP0
LP1    => LP1
```

Note that the file in position 2, JOHN.TXT, will now have two copies instead of three printed.



## SET QUEUE (Cont.)

### Notes

When you issue the PRINT command or the SUBMIT command (for batch processing), you specify attributes of the QMG job through command qualifiers. SET QUEUE command qualifiers change the matching attributes.

You cannot change the attributes of an active job.

You can delete files or change the number of copies to be printed by specifying /FILE\_POSITION:n in your SET QUEUE command line.

## STOP/ABORT

### 4.10.2.5 STOP/ABORT Command

The STOP/ABORT command 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

#### Parameter

printer

Specifies the line printer whose active job you want to stop.

#### Example

```
$ STOP/ABORT LPO 
```

This command stops the currently active job on LPO: This job is deleted from the queue and the next available job is queued.

#### Notes

As soon as the active job is deleted, the Queue Manager passes the eligible job to the processor. The processor has not been aborted or killed, only the active job on that processor.

**HOLD****4.10.2.6 HOLD**

HOLD blocks a job in its queue until it is explicitly released.

**Format**

HOLD/JOB queueName jobName

HOLD/ENTRY:nnn

**Notes**

Specifying the job's entry number saves you keystrokes. When you reference a job by its unique job entry number, you are avoiding ambiguity among job names. The entry number identifies both the queue the job is in and its name. If you supply the entry number, do not name the queue or the job.

Remember, jobs may share the same name but never the same entry number.

If you name the queue and job, do not supply the entry number.

When the system crashes, the Queue Manager puts all jobs in the queues in a HELD state. When the system is restored, you have to manually release those jobs using the RELEASE command.

## RELEASE

### 4.10.2.7 RELEASE

RELEASE unblocks a job that has been held in queue.

#### Format

```
RELEASE/JOB queueName jobName
```

```
RELEASE/ENTRY:nnn
```

#### Examples

```
$ PRINT/HOLD/NAME:MARY MUSE.TXT (RET)
```

The SHOW QUEUE/FULL command output might look like this:

```
** PRINT QUEUES **
PRINT => LP0
  [303,5]   MARY      ENTRY:698      FORM:0      HELD
          PRI:50   LEN:0      PAGE:0   NORESTART   FLAG:JOB   NOLOWER
COP:1
          > 1 DU0:[AMUSED]MUSE.TXT;1      COP:1      NODELETE
LP0      => LP0
LP1      => LP1
```

To unblock MUSE.TXT, you would type one of the following RELEASE commands on your terminal:

```
$ RELEASE/ENTRY:698 (RET)
```

```
$ RELEASE/JOB PRINT MARY (RET)
```

The SHOW QUEUE/FULL command output would look like this:

```
** PRINT QUEUES **
PRINT => LP0
  [303,5]   MARY      ENTRY:698      FORM:0      ACTIVE ON LP0
          PRI:50   LEN:0      PAGE:0   NORESTART   FLAG:JOB   NOLOWER
COP:1
          > 1 DU0:[AMUSED]MUSE.TXT;1      COP:1      NODELETE
LP0      => LP0
LP1      => LP1
```

#### Notes

The HELD status of the job in the previous example was first changed to WAITING, and then, because there were no jobs in queue, became ACTIVE.

### 4.11 SET AND SHOW

You can type SET commands to establish your default device, directory, and User Identification Code, and SHOW commands to display what these defaults are. You can change the protection code for any existing file in your directory with the SET PROTECTION command. In addition,

## SET AND SHOW

you can establish your own default file protection code with SET PROTECTION/DEFAULT and use SHOW PROTECTION to display your default protection.

The SET FILE command permits you to move the end-of-file marker in a file, truncate files, and create entries in one directory that point to files in another directory.

## SET DEFAULT

### 4.11.1 SET DEFAULT

SET DEFAULT changes where the operating system looks for information from your current device and directory to those indicated in the command. SET DEFAULT establishes your default device or directory, or both. That is, when you do not indicate a device or a directory in a file specification or command, the operating system supplies these defaults.

The protection codes for the directory and its files control your activity in the directory. If you are a nonprivileged user, you have the directory and file access rights of the WORLD category, and---depending upon your User Identification Code (UIC)---the GROUP category. If you are a privileged user, you have the access rights of the SYSTEM category. Of course, if your UIC matches the directory owner's, then you have the privileges of the OWNER category.

Note that a privileged user can change his UIC with the SET UIC command. If you intend to create or alter files in a directory other than your own, it is good practice to change your UIC to match the directory owner's. See Section 4.11.4.

SET DEFAULT/[NO]NAMED\_DIRECTORY establishes whether or not you can specify named directories in the SET DEFAULT command. Also, the SET DEFAULT command works differently if you previously set your terminal's default to /NONAMED\_DIRECTORY. See the description of this qualifier.

#### Format

```
SET DEFAULT[/qualifier] default[:]
```

#### Command Qualifier

```
/[NO]NAMED_DIRECTORY
```

#### Parameter

```
default[:]
```

Specifies a device and/or a directory name or logical name to be used as the default device or a directory in file specifications.

If you specify a physical device name, terminate the device name with a colon.

If you specify a directory name, you must enclose it in the square brackets. (Note, if your terminal's default is /NONAMED\_DIRECTORY, the directory field accepts only numbered directories.)

If you specify a logical name, the logical name must translate to an existing physical device and/or directory. If the logical name is to translate to a physical device, be sure that when you make the logical name assignment, you terminate the physical device name with a colon. When you issue the SHOW DEFAULT command, the system displays the actual physical device, not the logical name.

## SET DEFAULT (Cont.)

## Command Qualifier

/[NO]NAMED\_DIRECTORY

Allows SET DEFAULT to accept either named or numbered directories. The format for a named directory is [directory], where directory is one to nine of the following characters: the 26 letters A through Z, and the numbers 0 through 9. Examples of named directories include [MAG] and [1POTATO2].

The format for a numbered directory is [g,m], where g and m are octal numbers from 1 through 377. Examples of numbered directories are [100,2] and [202,10].

If you issue SET DEFAULT/NONAMED DIRECTORY, you can only specify numbered directories in the SET DEFAULT command. Operating with /NONAMED DIRECTORY is offered for users of other systems in the RSX-11 family who need to maintain compatibility. This qualifier is the default.

When you specify /NONAMED DIRECTORY, SET DEFAULT functions as it previously did on older versions of RSX. In the old style, the following happened:

- You can no longer set your default to a named directory. However, you can still use named directories in commands.
- You refer to directories in the SET DEFAULT command using the numbered format [g,m].
- The SET DEFAULT command now changes the UIC, as well as the default directory and device, of privileged users. In this case, the UIC and the directory name are identical numbers.
- The SET DEFAULT command with no arguments now returns the nonprivileged user to the login defaults and returns the privileged user to the login device, but does not change the default directory.

## Examples

```
$ SET DEFAULT DU1:[MASKELYNE] (RET)
```

This example sets DU1: as the default device and [MASKELYNE] as the default directory. All subsequent commands default to this device and directory unless you explicitly state otherwise.

```
$ SET DEF DU1: (RET)
```

```
$ ASSIGN DU1: SY0: (RET)
```

The two commands in this example are equivalent. In each case, the default device becomes DU1:. When you set the default device, SET DEFAULT actually assigns SY0 (or SY0: for systems without extended logical name support) as a logical name for the device you want. SY0 (or SY0:) is the logical name that represents your default device.

**SET DEFAULT (Cont.)**

```

$ SHOW ASSIGNMENTS (RET)
SYS$LOGIN = DU1:[TASK] (Login, Final, TT2: )

$ SET DEFAULT DU2:[IMAGE] (RET)
$ SHOW DEFAULT (RET)
DU2:[IMAGE]      Named      TT2:
Protection UIC:  [302,5]

$ SET DEFAULT SYS$LOGIN (RET)
$ SHOW DEFAULT (RET)
DU1:[TASK]      Named      TT2:
Protection UIC: [302,5]

```

This example shows the use of both explicit device and directory specifications (DU2:[IMAGE]) and logical names (SYS\$LOGIN, which is assigned to DU1:[TASK]) in the SET DEFAULT command line.

```

$ SHOW DEFAULT (RET)
DU0:[DONVAN]     Named      TT2:
Protection UIC:  [303,1]

$ SET DEFAULT/NONAMED DIRECTORY (RET)
$ SHOW DEFAULT (RET)
DU0:[303,1]      Nonamed    TT2:
Protection UIC:  [303,1]

$ SET DEFAULT [TINA] (RET)
SET -- Illegal directory

$ SET DEFAULT/NAMED_DIRECTORY [100,100] (RET)
$ SHOW DEFAULT go
DU0:[100,100]    Named      TT2:
Protection UIC:  [303,1]

$ SET DEFAULT/NONAMED DIRECTORY (RET)
$ SHOW DEFAULT FRET
DU0:[303,1]      Nonamed    TT2:
Protection UIC:  [303,1]

$ SET DEFAULT/NAMED_DIRECTORY [DONVAN] (RET)
$ SHOW DEFAULT (RET)
DU0:[DONVAN]     Named      TT2:
Protection UIC:  [303,1]

```

This example shows the behavior of SET DEFAULT/NONAMED\_DIRECTORY. You log in and issue a SHOW DEFAULT command. Your default directory is [DONVAN].

You issue SET DEFAULT/NONAMED\_DIRECTORY. You then type another SHOW DEFAULT command. The default directory appears as [303,1]. This is a directory name that is based on your User Identification Code (UIC). Note that [303,1] is an actual directory only if it had been previously created. Any time that you change your terminal's default from /NAMED\_DIRECTORY to /NONAMED\_DIRECTORY, the name of your default directory is the same as your UIC.



**SET DEFAULT (Cont.)**

You issue SET DEFAULT [TINA]. This command fails because your default is still /NONAMED\_DIRECTORY.

You issue SET DEFAULT/NAMED\_DIRECTORY [100,100]. This command is legal, because /NAMED\_DIRECTORY accepts both named and numbered directories. Doing a SHOW DEFAULT shows you that [100,100] is your default directory and that [303,1] is your protection UIC. You then switch to /NONAMED\_DIRECTORY and issue the SHOW DEFAULT command. Your default directory again is [303,1], a directory name based on your UIC.

Finally, you issue SET DEFAULT/NAMED\_DIRECTORY [DONVAN], thereby returning to your original login assignments.

**Notes**

On a privately allocated device, the operating system may assign different User Identification Codes to your files, depending on whether you create them in a named or a numbered directory. If you set your default to a named directory, such as [ADDRESS], any file you create in that directory is owned by your own UIC. However, if you set your default to a numbered directory, such as [1,1], any file that you create in that directory is owned by UIC [1,1]. This difference in file ownership, and therefore in file protection, does not adversely affect you. You nevertheless have full access to all files on a privately allocated device.

If your default on a privately allocated device is a named directory, you have the option of changing the UIC that owns the files created there. Use the SET UIC command, which is privileged.

Some application programs do not recognize named directories. If a task that you want to run fails for this reason, you can try the following solutions:

- Rebuild your task using the most recent version FCS or RMS-11, supplied with RSX-11M-PLUS. Note that you only need to rebuild your task either if the task contains FCS or RMS-11 routines or if the task uses FCSFSL to build the task file. If your task uses either FCSRES or RMSRES, you do not need to rebuild your task.

If rebuilding the task is not possible, try the next recommendation.

- Use either the COPY or RENAME commands to place the files that the task uses in a numbered directory. Note that only a privileged user, such as the system manager, can create directories on a public volume.

Next, use the SET DEFAULT command to make the numbered directory your default. Finally, run the task.

## SHOW DEFAULT


### 4.11.2 SHOW DEFAULT

SHOW DEFAULT displays the current default device and directory for your terminal. It also shows whether you are operating with /NAMED DIRECTORY or /NONAMED DIRECTORY as your default, and your User Identification Code (UIC).

#### Format

```
SHOW DEFAULT
```

#### Example

```
$ SHOW DEFAULT   
DU1: [HUSKY]   Named   TT6:  
Protection UIC: [15,20]
```

This example shows the display from SHOW DEFAULT, giving the default device, DU1:, and directory, [HUSKY], for your terminal, TT6:. Named indicates that your default is /NAMED\_DIRECTORY---that is, you can specify either a named or numbered directory in all file specifications and commands. Your User Identification Code (UIC) identifies you to the operating system and controls your ability to access files.

## SET FILE

## 4.11.3 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 Qualifier

```
/ENTER:synonym_filespec  
/NOWARNINGS  
/REMOVE  
/TRUNCATE
```

```
/END_OF_FILE:(BLOCK:n, BYTE:n)
```

## Parameter

```
filespec[s]
```

Identifies the file or files 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 `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 that file.

If you specify more than one file with this qualifier, the `synonym_filespec` 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 only be found with the VFY utility. Be careful.

## SET FILE (Cont.)

### /TRUNCATE

Enables you to get back unused disk space. When RSX-11M-PLUS 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 files.

Issue the command DIRECTORY/FULL 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 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 the extra allocated but unused blocks may speed up execution of a task.

### File Qualifier

#### /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 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, the pointer can be placed in the first byte of the next block. Say the number of blocks allocated is 22. In this case, you would specify (BLOCK:23, BYTE:0). The block number can be either octal or decimal.

For BYTE, the argument n specifies the location of the first unused byte of the specified block. The byte number can be octal or decimal. The maximum value for byte is 777 (octal), or 511 (decimal).

## SET FILE (Cont.)

## Examples

```
$ DIRECTORY/FULL HAI.FAI (RET)

DIRECTORY DU1: [HWANG]
16-JUN-85 11:02

HAI.FAI;1      (10,10)      0./55.
16-JUN-85 11:02
[222,314][RWED,RWED,RWED,R]

Total of 0./55. blocks in 1. files
$ SET FILE/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 qualifier, the user finds that while 55 blocks are allocated, none have been used. The file may contain nothing. It may contain irrelevant information, or it may contain useful data. In any case, the end-of-file pointer can be moved out to just past the last block allocated to check. 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 file. This makes it possible to examine the contents of the 55 allocated blocks.

```
$ SET FILE/ENTER:DUO:[MKTG]LANE.TXT ODIN.TXT (RET)
```

In this example, the user caused the file ODIN.TXT to appear in the [MKTG] 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 [MKTG]LANE.TXT (RET)
```

In this example, the user decided that the directory [MKTG] no longer needed access to 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. This saves disk space.

## SET UIC

### 4.11.4 SET UIC

SET UIC works differently depending on if your terminal's default is /NAMED\_DIRECTORY or /NONAMED\_DIRECTORY. If your terminal is /NONAMED\_DIRECTORY, see the Notes. If your terminal's default is /NAMED\_DIRECTORY, SET UIC is a privileged command that changes your User Identification Code (UIC). A UIC is an attribute of a directory and its files, establishing ownership. If you create or alter files in a directory other than your own, you can use this command to change your UIC to that of the directory's owner.

A UIC controls the protection codes of both a directory and its files. These protection codes determine who can access the directory and its files, and in what ways. See Section 4.6.1 for a description of how UICs relate to file protection codes.

If you change directories, but do not change your UIC to match that of the directory's owner, the following problems can occur:

- If you create a file in the directory of a nonprivileged account---that is, in a directory with a UIC group number greater than 10---the owner of that directory cannot easily change or delete that file. Most file protection codes restrict access in the WORLD category to READ, which is the operating system default for this category. Because your UIC owns the file, the owner of the directory may fall into the WORLD category, and therefore, can only read the file.
- Several system command files, as well as the command files of some users, set the UIC of the file's owner to match the UIC of the directory's owner. This matching of UICs gives the command file full access to all the files in the directory. If the files in that directory are owned by a different UIC, the command file may fail.

Note that if your terminal's default is /NAMED\_DIRECTORY, SET UIC does not change your directory. Typically, after executing this command, you issue the SET DEFAULT command to change your directory to the one belonging to the owner of the new UIC.

SET UIC works differently when your terminal's default is /NONAMED\_DIRECTORY. See the Notes.

#### Format

```
SET UIC [uic]
```

#### Parameter

[uic]

Specifies the User Identification Code (UIC) that you want to make your default. The format of a UIC is [g,m], where the first number is the group number, and the second is the member number. Group and member numbers are octal and range from 1 through 377.

Group numbers indicate privilege status. The UICs of privileged accounts have group numbers of 10 or less.

## SET UIC (Cont.)

## Examples

```

$ DIRECTORY/FULL [000000]GIDEON.DIR (RET)
Directory DU0:[000000]
12-JUN-85 14:12

GIDEON.DIR;1          (576,5040)      1./1.      C 12-JUN-85 07:28
  [200,1]  [RWED, RWED, RWED, R]

Total of 1./1. blocks in 1. file

$ SET UIC (RET)
UIC? [200,1] (RET)

$ SET DEFAULT [GIDEON] (RET)

```

Assume that, as a privileged user, you want to create or alter files in [GIDEON], another directory located on your current device. A full listing of the directory file itself, located in [000000]GIDEON.DIR, shows that the UIC of the directory's owner is [200,1]. You then issue the SET UIC command to change your current UIC to that of the directory's owner. Finally, you type the SET DEFAULT command to move to the other directory.

## Notes

As a privileged user, you can manipulate the files of another directory in any way. Note that you only need to issue the SET UIC command if you intend to create or alter files (WRITE) in another directory. However, when you are working with the files of another directory, it is good practice to always use this command.

If you change your UIC to that of a non-privileged account, you can change back to a privileged account by issuing the SET UIC command again.

You can type either the SET UIC or the SET DEFAULT command first; the sequence does not matter.

When a terminal's default is /NONAMED DIRECTORY, SET UIC changes both the UIC and the directory of a privileged user. For a privileged user, therefore, SET UIC functions exactly like SET DEFAULT.

## SHOW UIC

### 4.11.5 SHOW UIC

SHOW UIC displays your User Identification Code (UIC). Your UIC is unique and identifies you to the operating system. In addition, your UIC determines whether you are a privileged or nonprivileged user.

Each file and directory has a UIC associated with it, which identifies the owner. Each file and directory also has a protection code. The way in which your UIC relates to both the UIC and protection code of a file or directory controls your ability to access that file or directory. See Section 4.6 for a description of protection.

Nonprivileged users always have the same UIC, but privileged users can change their UIC with the SET UIC command.

#### Format

```
SHOW UIC
```

#### Example

```
$ SHOW UIC (RET)
Protection UIC: [303,17]
```

This example shows the display from SHOW UIC. The first number is the group number, and the second is the member number. Group and member numbers are octal and range from 1 through 377.

Generally, all users working on a particular project have the same group number. Each user account, however, has a unique member number.

Group numbers indicate privilege status. The UICs of privileged accounts have group numbers of 10 or less. The user issuing this SHOW UIC command, therefore, is nonprivileged.



## SET PROTECTION

## 4.11.6 SET PROTECTION

SET PROTECTION changes the protection code of an existing file or files.

The protection code determines which users may access a file and in what ways.

Nonprivileged users can change the protection of any files in their own directory. Privileged users can change 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 you are changing the protection. Multiple file specifications, separated by commas, and wildcards are permitted.

(code)

Specifies which user categories can access a file and what each user category may do to the file. The parentheses are required.

There are four categories of users in a file protection code, as follows:

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 used when the file was created.
GROUP	All users with the same UIC group number as the one the file was created under.
WORLD	All other users.

## SET PROTECTION (Cont.)

There are also four kinds of access to files, as follows:

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 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 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 the system default protection, unless you change their protection codes using the SET PROTECTION command, or establish an alternative default protection using the SET PROTECTION/DEFAULT command.

You specify the code in the SET PROTECTION command using the same format as shown in the previous example. Note, however, that you need to name only the user groups whose access rights you want to change and the types of access you want to grant to those groups. If you want to deny all access to a group, simply name the group and omit the colon (:) and the code for the access form.

### Command Qualifiers

/DATE:dd-mmm-yy

Specifies that you want to set the protection only on files created on the date given.

/SINCE:dd-mmm-yy

Specifies that you want to set the protection only on files created on or since the date given.

/THROUGH:dd-mmm-yy

Specifies that you want to set the protection only on files created on or before the date given.

## SET PROTECTION (Cont.)

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

Specifies that you want to set the protection only on files created within that range.

/TODAY

Specifies that you want to set the protection only on files created on the same day the command is being issued.

/EXCLUDE:filespec

Specifies that you want the SET PROTECTION command not 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 with a wildcard (\*).

## Examples

```
$ SET (RET)
FUNCTION?PROTECTION (RET)
CODE? (SYSTEM:RWED,OWNER:RWED,GROUP,WORLD) (RET)
File(s)? SANTONE.TEX (RET)

$ SHOW PROTECTION SANTONE.TEX (RET)
System:RWED, Owner:RWED, Group:No Access, World:No Access
```

This example changes the protection of file SANTONE.TEX so that privileged users and the owner have full access to the file and all others have no access to the file. The SHOW PROTECTION command displays the protection of file. Issuing the SHOW PROTECTION command is a good way of checking that you have changed the file's protection code correctly.

```
$ 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.

## 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 from inadvertently accessing your files, but you cannot prevent them from deliberately changing the files' protection status and doing as they want.

If you deny READ access to a task image file, the task cannot run.

Other commands accepting a protection code include INITIALIZE, INITIALIZE/UPDATE, MOUNT, and CREATE/DIRECTORY.

## SET PROTECTION (Cont.)

Other commands with qualifiers associated with file ownership include the following:

- INITIALIZE/OWNER\_UIC, see Chapter 5.
- INITIALIZE/UPDATE/OWNER\_UIC, see Chapter 5.
- INSTALL/UIC, see Chapter 7.
- LINK/OPTION:UIC, see Chapter 6, and the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.
- MOUNT/OWNER, see Chapter 5.
- RUN/UIC, see Chapter 7.
- SHOW QUEUE/OWNER\_UIC, see Chapter 4.
- SHOW TASK/DYNAMIC/OWNER, see Chapter 8.

## SET PROTECTION/DEFAULT

### 4.11.7 SET PROTECTION/[NO]DEFAULT

SET PROTECTION/DEFAULT establishes your personal default protection code for all files that you create after issuing this command. A file protection code controls the types of access that other system users have to your files.

Note that this command establishes your default file protection only for the current terminal session. To set your default file protection for all future terminal sessions, place the SET PROTECTION/DEFAULT command in your LOGIN.CMD file.

SET PROTECTION/NODEFAULT removes your personal default file protection. After issuing this command, the files you create receive the volume default protection. Although a volume can have any default file protection, this default is usually the same as the system default of (SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:R).

To change the protection code for existing files, see the description of SET PROTECTION in Section 4.11.6.

#### Format

```
SET PROTECTION:(code)/DEFAULT
```

```
SET PROTECTION/NODEFAULT
```

```
SET PROTECTION/DEFAULT
Code? (code)
```

#### Parameter

(code)

Specifies which user categories are permitted access to the file and what each user category may do to the file. The parentheses are required.

The file protection code has four user categories and four types of access allowed for each user category. See Section 4.11.6 for a description of the protection code format.

The first time you issue SET PROTECTION/DEFAULT in a terminal session, you must specify all four user categories. If you want to deny access to a category, omit the colon and the access code after the category name.

If you issue SET PROTECTION/DEFAULT again during the terminal session, you only need to specify the categories for which you want to change the access code. When you omit a user category in the protection code, you keep the current default protection for that category.

## SET PROTECTION/DEFAULT (Cont.)

### Examples

```
$ SET PROTECTION/DEFAULT (RET)
CODE?(SYSTEM: R,OWNER:RWED,GROUP:R,WORLD) (RET)
```

This example establishes your default file protection. Because you are issuing this command for the first time in the terminal session, you specify all four user categories. The code limits users in the SYSTEM and GROUP categories to READ access. You retain the standard volume default protection for the OWNER category of READ, WRITE, EXTEND, and DELETE. You deny all file access to the WORLD category by omitting the colon and access code after the category name.

```
$ SET PROTECTION:(GROUP) /DEFAULT (RET)
```

Assume that you type this command later in the same terminal session that you executed the command in the first example. Here you change your default protection for the GROUP category, denying these users any access to your files. Your default protection for the other three user categories remains unchanged. Note that you only needed to specify the user category for which you wanted to change the file access.

```
$ SET PROTECTION/NODEFAULT (RET)
```

This command removes your personal default file protection. The files that you create after issuing this command receive the volume default protection.

### Notes

If you place SET PROTECTION/DEFAULT in your LOGIN.CMD file, you must specify all four user categories.

The system manager can use the Account File Maintenance Program (ACNT) to establish a default file protection for your account. This method has the same effect as placing SET PROTECTION/DEFAULT in your LOGIN.CMD file: both methods establish your default file protection for each terminal session. Note, however, that you can modify your default file protection for any portion of a terminal session by typing the SET PROTECTION/[NO]DEFAULT command.

If both SET PROTECTION/DEFAULT and ACNT set your default file protection, then the protection indicated in SET PROTECTION/DEFAULT is your default.

SET PROTECTION/NODEFAULT removes your personal default protection, regardless of whether SET PROTECTION/DEFAULT or ACNT established it.

### Error Messages

Illegal user default protection code

**Explanation:** You did not specify all four user categories when issuing SET PROTECTION/DEFAULT the first time during a terminal session.

**User Action:** Reenter the command line, specifying all four user categories in the protection code.

## SET PROTECTION/DEFAULT (Cont.)

No protection specified for any field

**Explanation:** You attempted to set your default file protection code to deny file access to all four user categories---that is, you specified a file protection code that looks like (SYSTEM,OWNER,GROUP,WORLD).

**User Action:** The minimum default file protection code is READ access for one user category. Reenter the command line, specifying at least the minimum file access.

## SHOW PROTECTION

### 4.11.8 SHOW PROTECTION

SHOW PROTECTION displays your personal default file protection code.

Your default file protection can be established in two ways: either by issuing the SET PROTECTION/DEFAULT command, or by using the Account File Maintenance Program (ACNT) to enter a protection code for your account. See the description of SET PROTECTION/DEFAULT in Section 4.11.7.


If you do not set your own default file protection, then SHOW PROTECTION issues the following message:

No user default protection specified.

#### Format

SHOW PROTECTION

#### Example

```
$ SHOW PROTECTION   
System:RWED, Owner:RWED, Group:R, World:No Access
```

This example shows the display from SHOW PROTECTION, indicating that your default file protection grants full access to the SYSTEM and OWNER categories, READ access to the GROUP category, and no access to the WORLD category.



## 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 to 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 these devices.

In RSX-11M-PLUS 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 introduces the important concepts concerning devices and volumes. In addition, the chapter describes the DCL commands that control devices and volumes.

#### 5.1 DEVICES ON RSX-11M-PLUS SYSTEMS

Table 5-1 lists the devices found on RSX-11M-PLUS systems. Each installation has a different configuration 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-PLUS 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 want to specify DU0: in a command, DU: will do.

# HANDLING DEVICES AND VOLUMES

Table 5-1  
Devices on RSX-11M-PLUS Systems

Mnemonic	Device
BS:	DT07 UNIBUS switch
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/J/F disk drive
DL:	RL01/RL02 disk drives
DM:	RK06/RK07 disk drive
DR:	RM02/RM03/RM05 disk drives and RM80/RP07 fixed disk drives
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 drives
DY:	RX02 diskette drives
EM:	Non-Rotating Electronic Memory (ML11)
II:	Interprocessor Interface
IP:	IP11 Industrial Control System
LA:	Laboratory Peripheral Accelerator
LB:	System Default Pseudo Device
LP:	Line Printer
LR:	PCL-11 Parallel Communications Link (XCUR)
LT:	PCL-11 Parallel Communications Link (XMTT)
MK:	MKA11 Programmable Memory Box
MM:	TU16/TU45/TE16/TU77 magnetic tape drives

(continued on next page)

## HANDLING DEVICES AND VOLUMES

Table 5-1 (Cont.)  
Devices on RSX-11M-PLUS Systems

Mnemonic	Device
MS:	TS11/TS05/TK25 tape drives
MT:	TS03/TU10/TE10 magnetic tape drives
MU:	TK50 tape drive
NL:	Null Device
PP:	Paper Tape Punch
PR:	Paper Tape Reader
RD:	Reconfiguration Driver
SP:	Spooling Pseudo Device
SY:	User Default Pseudo Device
TI:	Terminal Input Pseudo Device
TT:	Terminal
VT:	Virtual Terminal
WK:	Workfile Pseudo Device
XM:	DMC-11 Interprocessor Link

RSX-11M-PLUS 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 the RL02 disk), but you should remember that the driver may support more than one device. For instance, the DU: driver supports the RD51 and RD52 fixed disks and the RX50 diskette.

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-PLUS include the paper tape punch (PP:), paper tape reader (PR:), the TU60 cassette (CT:), and the 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.

## HANDLING DEVICES AND VOLUMES

On RSX-11M-PLUS 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-PLUS systems. Files-11 volumes are those on which the data is organized according to the normal file structure used by RSX-11M-PLUS 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 5.11). 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 5.9) 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. You can use unlabeled tapes for backup on RSX-11M-PLUS. 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 and DIRECTORY).

Magnetic tapes are supported in two ways: file structured and foreign. Mounting a tape foreign allows the tape to be processed efficiently and in nonstandard ways. The most common example of this type of processing occurs during backup. Backup or restore operations using tape require that the tape be mounted using the /FOREIGN qualifier. Mounting a file structured tape (the default) means that the tape is in a Files-11 format and that the tape is to be processed in a standard manner. Most operations (for example, COPY and DIRECTORY) require that the tape be mounted as a Files-11 volume.

RSX-11M-PLUS includes the magnetic tape ACP (MTAACP), which allows you to use labeled tapes. Labeled magnetic tape volumes conform to the American National Standard for Magnetic Tape Labels and File Structure for Information Exchange (ANSI X3.27-1978). Labeled magnetic tapes are sometimes called Files-11 magnetic tapes, but this is an inaccurate term; the better term is labeled or ANSI magnetic tapes. You can create labeled magnetic tape volumes through the INITIALIZE command (Section 5.11). The default when mounting tapes is labeled; the tape is assumed to be ANSI format.

The MTAACP also allows you to access tapes that have non-ANSI format labels, or no labels at all. Mounting the tape with the /NOLABEL qualifier makes such access possible, allowing you to access tapes that are not in ANSI format from high-level languages.

## HANDLING DEVICES AND VOLUMES

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:, IP:, IS:)
- Laboratory devices (LA:, LS:)
- Universal digital controller (UD:)

The system uses several physical devices in its operations. These devices include the following:

- Communications devices, including parallel communications links (LR:, LT:)
- Interprocessor links (XB:, 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 denotes the volume. For instance, if you have files on a volume to which you want to have private access, you take the following steps:

1. Use the SHOW DEVICES dd: command to see which devices of the type you want to use are available. The kind of magnetic medium the volume is on determines which type of device you can use.
2. Make the device your private device through the ALLOCATE command.
3. Physically place the magnetic medium on the device drive. (If necessary, your system operator can do this for you.)
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 cases, this distinction is not important, but you should understand that the expression ddnn: sometimes refers 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-PLUS systems offer you a number of ways to name and access physical devices. You can directly access physical devices by naming them in commands or file specifications. However, you can use several forms of indirect access to physical devices as well.

The operating system often accesses physical devices as pseudo devices. Tasks access physical devices through logical unit numbers (LUNs). LUNs establish the relationship between the I/O done by the task and the devices on the system that the task needs to access. 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 installed, 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. During one installation, this might be DUL:, while during another, this might be DL0:; but regardless of the actual physical device involved, the name LB: always refers to the current disk location of the system libraries.

Some pseudo device names are really only logical names. No system data structures are associated with logical names other than as 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.

The term pseudo device does not refer to a single form of software, but rather to the concept that a usable device name is unknown to the system regardless of the installation's device configuration. The pseudo device MK:, for example, requires the presence of certain hardware.

Table 5-2 summarizes the pseudo devices on RSX-11M-PLUS systems. Pseudo device CO: is handled differently depending on whether the system includes Console Logging. Without Console Logging, CO: is usually redirected to the main operator's terminal, which is often TT0:. Including Console Logging means a driver called CO: is used. This driver permits your system manager to send 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.

On RSX-11M-PLUS systems, the library directory is also on pseudo device LB:.

# HANDLING DEVICES AND VOLUMES

Table 5-2  
Pseudo Devices on RSX-11M-PLUS

Device Name	How Implemented	Comments
CO:	Redirection	No Console Logging
	Driver	Console Logging present
CL:	Redirection	
LB:	Redirection	
MK:	Driver	Requires hardware
NL:	Driver	Nonphysical device
RD:	Driver	Nonphysical device
SP:	Redirection	
SY:	Assignment or	User default device
	Redirection	
TI:	Dynamic	
	Redirection	
VT:	Driver	
WK:	Assignment	
dd:	Driver	Nonphysical device, requires hardware

Pseudo device MK: requires certain hardware to be present and is implemented through redirection for internal use on RSX-11M-PLUS systems.

Pseudo device NL:, the null device, is implemented through the NL: driver. The NL: pseudo device can be used for program testing that requires an available input or output device. 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 RD: is the Reconfiguration Driver.

Pseudo device SP: is used by the Queue Manager for spooling and is implemented through redirection.

The SY: device can be created in two ways. First, pseudo device SY: is always redirected to the device from which you bootstrap the system. Second, pseudo device SY: is also 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

## HANDLING DEVICES AND VOLUMES

Maintenance Program). ACNT names the default device for all commands that do not specify a device name. 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.

On RSX-11M-PLUS systems, the VT: driver is used for virtual terminals in batch processing.

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 5.4), redirect devices (see the ASSIGN/REDIRECT command, Section 5.4.3, and write drivers. (See the RSX-11M-PLUS Guide to Writing an I/O Driver.)

### 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 the physical device units that the task accesses. 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 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.
TI0:	5.
CL0:	6.



## HANDLING DEVICES AND VOLUMES

This means a task linked with the default LUN assignments can simultaneously refer 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 pseudo 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 hardcopy 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.

### 5.1.2.3 Logical Device Names

#### NOTE

This section applies only if your RSX-11M-PLUS operating system does not support extended logical names. If your RSX-11M-PLUS operating system has logical name support, please read Section 5.2 for more information. If you are not sure, see your system manager.

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-PLUS device or pseudo device, such as DU: 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 5.4.)

There are three types of logical device assignments:

1. Local assignments, which can be made by any user. Local assignments 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 logical name to different devices.

## HANDLING DEVICES AND VOLUMES

2. Certain login assignments (SY:, SYS\$LOGIN:), which can be controlled 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 global assignments (the next category).
  - 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, Shareable, Private, and Unowned Devices and Mounted Volumes

You must use the MOUNT command on RSX-11M-PLUS to access to 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, use of the MOUNT/FOREIGN command bypasses the ACP and the file system. (MOUNT/FOREIGN is required for foreign volumes on RSX-11M-PLUS.) Use MOUNT/FOREIGN to initialize volume or to use a custom ACP.

RSX-11M-PLUS systems support four categories of device ownership: 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 are accessible to no one. Volume protection and file protection can greatly alter your rights of access to a particular volume or file.

Following are descriptions of the four categories of device ownership:

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 5.13.2) or the /PUBLIC qualifier to the MOUNT command (Section 5.9). 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. 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 5.7), 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-PLUS permits no access to unowned volumes.

#### 5.1.4 Preparing 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. You must use a combination of several system utilities and DCL commands 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, using the DCL ALLOCATE command. Make sure the device is write-enabled.
2. After physically placing the disk on the drive, spin it up and mount it using the /FOREIGN qualifier. This permits the system to bypass the file system when accessing the disk.
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 questions about formatting a disk, see your system manager.

If formatting is necessary, you can use the Disk Volume Formatter Utility (FMT). (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 purpose. This information is usually written on disks when they are manufactured, but you can supplement this information with the ANALYZE/MEDIA command.
5. Once you have identified the bad blocks, the disk driver and the disk controller have full use of the disk. 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.

## HANDLING DEVICES AND VOLUMES

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 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.
BADBLK.SYS	This is the bad block file. All unusable blocks are included in this file. Any block that is included in this file is unavailable for any other use.
BITMAP.SYS	This file contains information on all used and free blocks 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 then mount it as a Files-11 volume. Use the following 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 INITIALIZE placed in the Volume Home Block 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 if you use the following DCL commands:

- BACKUP
- INITIALIZE

## HANDLING DEVICES AND VOLUMES

This requirement may also arise if you use the following utilities, described in the RSX-11M/M-PLUS Utilities Manual:

- FLX, File Transfer Utility
- FMT, Disk Volume Formatter Utility
- BAD, Bad Block Locator Utility, accessed from DCL through the ANALYZE/MEDIA command
- BRU, Backup and Restore Utility, accessed from DCL through the BACKUP command
- DSC, Disk Save and Compress Utility
- VFY, File Structure Verification Utility
- DMP, File Dump Utility

### 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-PLUS. 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-PLUS utilities and commands.

1. Allocate an appropriate device and place the tape on the drive. Make sure the drive is write-enabled.
2. After you physically place the tape on the drive, mount the tape using the /FOREIGN qualifier. This permits the system to bypass the file system when accessing the tape.
3. Next, you must provide information that the file system needs to use the tape. This is done through the INITIALIZE command. Qualifiers to the INITIALIZE command 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 then remount it as an ANSI volume, using the DISMOUNT command and the MOUNT command (this time without the /FOREIGN qualifier).

### 5.1.6 Summary of Device and Volume Use Commands

All Files-11 volumes, ANSI magnetic tapes, and unlabeled magnetic tapes must be mounted for any access.

All other volumes are considered foreign volumes. On RSX-11M-PLUS you must mount foreign volumes using the MOUNT/FOREIGN command.

## HANDLING DEVICES AND VOLUMES

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 three 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.
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.

Any user who mounts a volume has full access to that volume within the limits of privilege and the volume and file protection. Privileged users have privileged access and nonprivileged users have nonprivileged access. However, nonprivileged users do 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.

The LOGOUT command 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.

### 5.2 LOGICAL NAMES

#### NOTE

This section applies only if your RSX-11M-PLUS operating system supports extended logical names. If you are not sure, see your system manager.

A logical name is a user- or system-defined name for an equivalence name, which can be the following:

- All or part of a file specification - This keeps your programs and command procedures independent of physical file specifications.
- A physical device---You can assign logical names to devices such as magnetic tape drives, terminals, and line printers. The system manager can assign logical names to public disk volumes, so that users do not have to be concerned with the physical location of these volumes.
- Anything created by the DEFINE command (see Section 5.6 for more information).

In addition, to reduce typing, you can use logical names as a shorthand way of specifying files or directories that you refer to frequently. For example, you might assign the logical name HOME to

your default disk and directory, or the logical name DIARY to a file in which you keep a log of your daily activities. You can also use logical names in file specifications to keep your programs and command procedures independent of physical file specifications.

### 5.2.1 Logical Name Tables

The system maintains logical name and equivalence name pairs in four logical name tables:

1. Task logical name table---Contains logical name entries that are created for an individual task using the Create Logical (CLOG\$) directive. These entries remain in the table for only as long as the task is running. When either the task has completed execution or the Delete Logical (DLOG\$) directive has been issued, the logical names are removed from the table. (See the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual for more information on the CLOG\$ and the DLOG\$ directives )
2. User logical name table---Contains logical name entries that are local to a particular user. By default, the DEFINE and ASSIGN commands and login assignments place a logical name in the user logical name table.
3. Group logical name table---Contains logical name entries that are qualified by a group number. These entries can be accessed only by tasks that execute with the same group number in their user Identification Codes (UIC) as the user that assigned the logical name. You must use the /GROUP qualifier to make an entry in the group logical name table.
4. System logical name table---Contains entries that can be accessed by any task in the system. You must use the /SYSTEM or /GLOBAL qualifier to make an entry in the system logical name table.

You must be privileged to place entries in the group and system logical name tables.

#### 5.2.1.1 Displaying Logical Name Table Entries

The SHOW LOGICALS (or SHOW ASSIGNMENTS) command displays current entries in the logical name tables.

To display the contents of the user logical name table, enter the SHOW LOGICALS command, without any qualifiers or parameters as follows:

```
$ SHOW LOGICALS RET
```

This command produces a display of the current logical names in the user logical name table and their equivalence.

## HANDLING DEVICES AND VOLUMES

You can request the system to display all entries in the specific logical name table. For example:

```
$ SHOW LOGICALS/SYSTEM (or /GLOBAL)
    /GROUP:g
    /LOCAL (default)
    /LOGIN
    /ALL
    /TERMINAL:ttnn:
```

The SHOW LOGICALS/SYSTEM (or /GLOBAL) command displays all the logical assignments in the system, including global assignments.

The SHOW LOGICALS/GROUP command displays the logical names accessible to users with the same group number. The argument g allows you to specify a specific group number. (This is a privileged command.)

The SHOW LOGICALS/LOCAL command displays the local and login logical assignments.

The SHOW LOGICALS/LOGIN command displays only the login logical assignments.

The SHOW LOGICALS/ALL command displays the system, your group, local, and login logical assignments.

The SHOW LOGICALS/TERMINAL:ttnn: command displays local and login logical assignments for the specified terminal (ttnn:). The /TERMINAL qualifier may be used in conjunction with the /LOCAL, /LOGIN, or /ALL qualifiers.

See Section 5.13.1 for more information.

### 5.2.2 How To Create and Delete Logical Names

Logical names and equivalence name strings can each have a maximum of 79 characters, and can be used to form all or part of a file specification. If only part of a file specification is a logical name, it must be the leftmost component of the file specification. You can then specify the logical name in place of the device (or device and directory name) in subsequent file specifications, terminated by a colon (:).

When you specify an equivalence name for the ASSIGN command, you must specify it using the proper punctuation marks (colons, brackets, periods). If you specify only a device name, terminate the equivalence name parameter with a colon (:); if you specify a device and directory name, or a full file specification, do not terminate the equivalence name with a colon.

You can optionally terminate a logical name with a colon (:). If you do this, the ASSIGN command removes the colon before placing the logical name in the logical name table. The DEFINE command, on the other hand, does not remove the colon before placing the logical name in the logical name table. However, if you specify a colon at the end of the logical name, and if you want the colon as part of the logical name, you use the DEFINE command to create the logical.



To delete a logical name, use the DEASSIGN command. Generally, to delete a logical name created by the DEFINE command, you should put quotation marks (") on both sides of the logical name in the DEASSIGN command line. For example, to delete the logical name TASK: (created by the DEFINE command), use the following command line:

```
$ DEASSIGN "TASK:" (RET)
```

You do not need the quotation marks to delete any logical names created by the ASSIGN command.

### 5.2.3 Logical Name Translation

For logical names created using the ASSIGN command, if the system finds a logical name, it substitutes the equivalence name for the logical name in the file specification. This is called logical name translation.

When the system translates logical names, it searches the task, user, group, and system tables (in that order), and uses the first match it finds.

If you are ever in doubt about the equivalence name assigned to a logical name, use either the SHOW LOGICALS or the SHOW ASSIGNMENTS command.

#### 5.2.3.1 Iterative Translation

When the system translates logical names in file specifications, the logical name translation can be iterative. This means that after the system translates a logical name in a file specification, it repeats the process of translating the file specification. For example, consider the following logical name table entries made with the ASSIGN commands:

```
$ ASSIGN LB:[3,54] RANDOM (RET)
$ ASSIGN RANDOM:VMR.TSK TASK (RET)
```

The first ASSIGN command equates the logical name RANDOM to the device and directory specification LB: [3,54]. The second ASSIGN command equates the logical name TASK to the equivalence string RANDOM:VMR.TSK. In subsequent commands, or in programs you execute, you can refer to the logical name TASK. For example:

```
$ RUN TASK (RET)
```

When the system translates the logical name TASK, it finds the equivalence string RANDOM:VMR.TSK. The system then checks to see if the portion to the left of the colon (if there is a colon) in the equivalence name is a logical name; if it is a logical name (as RANDOM is in this example), the system translates that logical name also. When the logical name translation is complete, the translated device and file specification is LB:[3,54]VMR.TSK.

#### NOTE

The system limits logical name translation to 10 levels. If you define more than 10 levels or create a circular definition, an error occurs when the logical name is used.

## HANDLING DEVICES AND VOLUMES

You can control the translation of logical names by using the /FINAL qualifier on either the ASSIGN or DEFINE commands. For example, you assign three levels of logical names in the following manner:

```
$ ASSIGN DB0:[TSTSYS] TESTS: (RET)
$ ASSIGN/FINAL TESTS: LAST: (RET)
$ ASSIGN LAST: DISK: (RET)
```

Without the /FINAL qualifier on the second assignment, the logical name DISK: would eventually be translated into the equivalence name DB0:[TSTSYS]. Because you applied the /FINAL qualifier to the second assignment, the translation stops at the equivalence name TESTS:. The task is less hardware dependent when you use this qualifier. If for some reason, DB0: is no longer available for use, you can assign TESTS: to point to another disk drive (when the task is not running) without having to change the task itself. However, at some point, you (or the task) have to refer to the logical name TESTS: (for example, spawning another task, issuing a ACHN\$ directive) in order for the translation from DISK: to DB0:[TSTSYS] to work. For more information on the ACHN\$ directive, see the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual.

**ANALYZE/MEDIA****5.3 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 are 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.

## ANALYZE/MEDIA (Cont.)

/[NO]EXERCISE[=(n,m)]

Specifies the double word data pattern 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 without testing them.

/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 operations 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 /BAD qualifier.

### Examples

```
$ ANALYZE/MEDIA (RET)
Device? DU1: (RET)
```

In this example, ANALYZE/MEDIA analyzes the media found in DU1: 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 your terminal.

```
$ ANALYZE/MEDIA/BADBLOCKS DU1: (RET)
BAD> LBN(S)= 300 (RET)
BAD> LBN(S)= 1002 (RET)
BAD> LBN(S)= (CTRL/Z)
BAD -- DU1: Total bad blocks = 2.
```

In this example, ANALYZE/MEDIA prompts you for bad block information to add to the bad block descriptor file of DU1:. After you enter all the bad blocks, enter 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 enter 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 (Cont.)

```

$ ANALYZE/MEDIA/BADBLOCKS/SHOWDUL: (RET)
BAD>LBN(S) = 3 (RET)
BAD>LBN(S) = 347 (RET)
BAD>LBN(S) = (CTL/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.

**Notes**

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 message:

```
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:** Bad 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 switch 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.

## ANALYZE/MEDIA (Cont.)

BAD---ddnn: Bad block file overflow

**Explanation:** BAD 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.

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 only applies 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:** BAD 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.

## ANALYZE/MEDIA (Cont.)

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 which was allocated to [0,0]BADBLK.SYS already existed in the bad block descriptor file.

**User Action:** Reenter the command line with another value.

BAD---ddnn: Failed to attach

**Explanation:** BAD 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 only applies 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:** BAD 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.

## ANALYZE/MEDIA (Cont.)

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.

BAD---ddnn: Home block not found

**Explanation:** BAD was unable to read the home block while attempting to validate the volume label. This message only applies 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-PLUS 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.



## ANALYZE/MEDIA (Cont.)

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:** BAD detected a syntax error on the command line.

**User Action:** Determine the correct syntax and reenter the command line.

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 bad blocks count on the physical volume label.

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

### 5.4 ASSIGN

ASSIGN can behave in two distinct ways, depending on whether your RSX-11M-PLUS operating system includes extended logical name support. For RSX-11M-PLUS systems without extended logical name support, read Section 5.4.1. For RSX-11M-PLUS systems with extended logical name support, read Section 5.4.2. If you are not sure, see your system manager.

#### 5.4.1 For RSX-11M-PLUS Systems Without Extended Logical Name Support

ASSIGN associates a logical name with a physical device, pseudo device, or another logical device. All references to the logical name are resolved by the operating system to the associated physical device, pseudo device, or logical device.

#### Format

```
ASSIGN
Device?   ddnn:
Logical device?  logicalddnn:

ASSIGN[/qualifier[s]] ddnn: logicalddnn:
```

#### Command Qualifiers

```
/GLOBAL
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:
```

#### 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 DU0: or SP0:, or random letters, such as AZ1:.

**ASSIGN (Cont.)****Command Qualifiers****/GLOBAL****/SYSTEM**

Specifies that the assignment is to be a system table assignment. **/SYSTEM** and **/GLOBAL** are synonyms and are privileged qualifiers. System 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 or files 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 own set of local assignments for your terminal. These local logical names exist only for your terminal.

**/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 to the system, the system automatically assigns the logical device name **SYO:** 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.

**/TERMINAL:ttnn:**

Specifies that the requested local assignment be applied to another terminal. Only a privileged user may make assignments to other terminals. Note that the target terminal must be logged in before the assignment can be made.

**Examples**

```
$ ASSIGN (RET)
Logical name equivalent?DU1: (RET)
Logical name? TP1: (RET)
```

This example assigns the logical name **TP1:** to the physical device **DU1:**. The user may now issue commands referring to device **TP1:** (in any command that accepts a device specification) and **DU1:** will be substituted for it.

## ASSIGN (Cont.)

\$ **ASSIGN LB0: RR2:** RET

This example assigns the logical name RR2: to the pseudo device LB0:. This logical name exists for your terminal only.

\$ **ASSIGN/TERMINAL:TT4: DY1: LN1:** RET

This example assigns the logical name LN1: to the physical device DY1: for all commands and tasks initiated from TT4:. This command must be issued from a privileged terminal. Also, the terminal TT4: must be logged in before the assignment can be made.

\$ **ASSIGN/GLOBAL DL1: XX1:** RET

This example assigns the logical name XX1: to the physical device DL1:. 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. DL1: can also be used to reference DL1:.

### 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.

You can display current assignments with either the SHOW ASSIGNMENTS.

You can assign SY0: with SET DEFAULT, described in Chapter 4.

ASSIGN/REDIRECT is a separate command, described in 5.4.3.

ASSIGN/TASK is a separate command.

ASSIGN/QUEUE is a separate command, described in the RSX-11M/M-PLUS Batch and Queue Operations Manual.

ASS---Device not in system

**Explanation:** The specified equivalence device name does not exist in the system.

**User Action:** Reenter the command line specifying a device that is recognized by the system.

ASS---Device not terminal

**Explanation:** You did not specify a terminal when you issued the ASSIGN/TERMINAL command.

**User Action:** Reenter the command line specifying a terminal.

**ASSIGN (Cont.)**

ASS---Terminal not logged in

**Explanation:** The terminal you attempted to make a logical assignment to is not logged in.

**User Action:** Log in the terminal you want to make the assignment for and reenter the command line.

#### 5.4.2 For RSX-11M-PLUS Systems With Extended Logical Name Support

ASSIGN equates a logical name to a Files-11 physical device name, to all or part of a Files-11 file specification, or to another logical name. All references to the logical name are resolved by the operating system.

#### Format

```
ASSIGN
Logical name equivalent?  equivalence_name
Logical name?  logical_name

ASSIGN[/qualifier[s]] equivalence_name logical_name
```

#### Command Qualifiers

```
/FINAL
/GLOBAL
/GROUP[:g]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:
/TRANSLATION_ATTRIBUTES=TERMINAL
```

#### Parameters

equivalence\_name

The Files-11 device or file specification that you have defined as the substitution for the logical name. The equivalence name can also be another logical name that will be iteratively invoked.

ASSIGN checks the syntax of an equivalence name that is a device or file specification. When you specify an equivalence name that will be used as a file specification, you must include the punctuation marks (colons, brackets, periods) that would be required if the equivalence name were used directly as a file specification. Therefore, if you specify a device name as an equivalence name, terminate the device name with a colon. ASSIGN will not remove the terminating colon. If you did not correctly specify the device or file, the ASSIGN command fails. Also, if you specify quotation marks (") around the equivalence name, ASSIGN retains the quotation marks. (This differs from the DEFINE command, which strips the quotation marks.)

## ASSIGN (Cont.)

### logical\_name

The name you selected that you want to give the device or file specification.

If you terminate the logical name with one colon (:) or two colons (::), the system removes the colons before placing the name in the logical name table. (This differs from the DEFINE command, which saves the colons.) ASSIGN removes the colons because FCS and RMS (file access methods) do not consider the terminating colons to be part of the logical name that either file access method attempts to process.

A logical name may contain the following ASCII characters: the 26 letters A through Z, the numbers 0 through 9, and the characters underscore (\_), colon (:), and dollar sign (\$).

The length of the logical name is limited by the legal length of a command line, which is 79 characters. For example, if the length of ASSIGN plus qualifiers plus the equivalence name plus spaces is 24 characters, the logical name cannot be more than 55 characters long.

ASSIGN will do complete logical translation of the equivalence string and create the logical with the /FINAL qualifier. The logical name and equivalence name consist of a device in the DDnn: format. The physical device specified in the equivalence name must exist in the system. These guidelines provide compatibility with previous logical name capability.

### Command Qualifiers

/GLOBAL  
/SYSTEM

Specifies that the assignment is to be a system table assignment. /SYSTEM and /GLOBAL are synonyms and are privileged qualifiers. System assignments apply to all tasks running in the system.

/GROUP[:g]

Specifies that the assignment is to be a group assignment.

The argument g is the User Identification Code (UIC) group number of the users who share the logical name. If you do not specify a group number, the default is your own group number, which is taken from your current protection UIC. If you are nonprivileged, the group number is same as the UIC that is assigned to you when you log in.

/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 or files 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.

## ASSIGN (Cont.)

You can define your own set of local assignments' for your terminal. These local logical names exist only for your terminal.

/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 to the system, the system automatically assigns the logical name SYS\$LOGIN to the user's default device and directory, which is the device and directory that contains the user's files.

/TERMINAL:ttnn:

Specifies that the requested local assignment be applied to another terminal. Only a privileged user may make assignments to other terminals. Note that the target terminal must be logged in before the assignment can be made.

/FINAL

/TRANSLATION\_ATTRIBUTES=TERMINAL

Specifies that the equivalence name string should not be translated iteratively; that is, the logical name translation should terminate with the current equivalence string. See Section 5.2.3.1 for more information on the iterative translation of logical names. The /TRANSLATION\_ATTRIBUTES=TERMINAL qualifier is a synonym for /FINAL, included for VAX/VMS compatibility.

### Examples

```
$ ASSIGN (RET)
Logical name equivalent? DU1: (RET)
Logical name? TP1 (RET)
```

This example assigns the logical name TP1 to the physical device DU1:. The user may now issue commands referring to device TP1 (in any command that accepts a device specification) and DU1: will be substituted for it.

```
$ ASSIGN LBO: RR2: (RET)
```

This example assigns the logical name RR2 to the pseudo device LBO:. This logical name exists for your terminal only.

```
$ ASSIGN/GROUP:303 DU0:[1,1]SYSLIB.OLB;3 SYS$LIB (RET)
```

This example assigns the logical name SYS\$LIB to file SYSLIB.OLB;3, which is located on device DU0: in directory [1,1]. This logical name exists for users whose User Identification Code (UIC) group number is 303.

**ASSIGN (Cont.)**

```
$ ASSIGN/TERMINAL:TT4: DU2:[TEST] A: (RET)
$ SHOW ASSIGNMENTS/TERMINAL:TT4: (RET)
A = DU2:[TEST] (Local, TT4:)
```

This example assigns the logical name A to the directory [TEST], which is located on device DU2: for all commands and tasks initiated from TT4:. The user then issues a SHOW ASSIGNMENTS to display the logical name. These commands must be issued from a privileged terminal. Also, the terminal TT4: must be logged in before the assignment can be made.

```
$ ASSIGN/GLOBAL DL1: XX1: (RET)
```

This example assigns the logical name XX1 to the physical device DL1:. 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. The string DL1: can also be used to reference the physical device DL1:.

```
$ ASSIGN DU1 DQ (RET)
$ SHOW ASSIGNMENTS (RET)
DQ = DU1: (Local, TT:)
```

This example assigns the logical name DQ to the physical device DU1:. Although the user did not terminate the equivalence name (DU1) with a colon, DCL recognizes that the equivalence name resembles the other ASSIGN command device format (ddnnn:, as described in Section 5.4.1). Because the ASSIGN command line is in this format, DCL automatically terminates the equivalence name string with a colon, as shown when the user issued the SHOW ASSIGNMENTS command.

```
$ ASSIGN DB0: [TSTSYS] TESTS (RET)
$ ASSIGN/FINAL TESTS LAST: (RET)
$ ASSIGNLAST FIRST (RET)
```

This example shows three assignments. First, the user assigns logical name TESTS to the equivalent name DB0:[TSTSYS]. Second, the user then assigns the logical name LAST to the equivalent name TEST5 with the /FINAL qualifier. Third, the user assigns the logical name FIRST to the equivalent name LAST. When a task (for example) refers to the logical name FIRST, the translation is carried only as far as the equivalent name TESTS and then stops. For more information on the iterative translations of logical names, see Section 5.2.3.1.



**ASSIGN (Cont.)****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 name DEV has a global assignment of DB1: but a local assignment of DR0:, the operating system interprets DEV to be DR0: for your terminal.

ASSIGN is counteracted by DEASSIGN.

You can display current assignments with either the SHOW ASSIGNMENTS or SHOW LOGICALS command.

The logical name SY0: can be assigned only to a device.

ASSIGN/REDIRECT is a separate command, described in 5.4.3.

ASSIGN/TASK is a separate command.

ASSIGN/QUEUE is a separate command, described in the RSX-11M/M-PLUS Batch and Queue Operations Manual.

ASS---Device not in system

**Explanation:** The specified equivalence device name does not exist in the system.

**User Action:** Reenter the command line specifying a device that is recognized by the system.

ASS---Device not terminal

**Explanation:** You did not specify a terminal when you issued the ASSIGN/TERMINAL command.

**User Action:** Reenter the command line specifying a terminal.

ASS---OCTAL group number expected

**Explanation:** You did not specify an octal group number value when you issued the DEFINE/GROUP command.

**User Action:** Check the group number that you want and reenter the command.

ASS---Terminal not logged in

**Explanation:** The terminal you attempted to make a logical assignment to is not logged in.

**User Action:** Log in the terminal you want to make the assignment for and reenter the command line.

## ASSIGN/REDIRECT

### 5.4.3 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 you have another device that can take the place of a currently unavailable device. The devices need not be of the same type.

#### Format

```

ASSIGN/REDIRECT
From?  oldddnn:
To?    newddnn:

ASSIGN/REDIRECT oldddnn: newddnn:

```

#### Parameters

oldddnn:

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 DU0: DU3: (RET)
$ SHOW DEVICES TYPE DU: (RET)
DU0:          DU3:          LOADED
DU1:          LOADED
DU2:          LOADED
DU3:          LOADED

$ ASSIGN/REDIRECT (RET)
FROM? DU0: (RET)
TO?   DU0: (RET)
$ SHOW DEV DU: (RET)
DU0:          LOADED
DU1:          LOADED
DU2:          LOADED
DU3:          LOADED

```

This example shows the process of redirection. Following the first ASSIGN/REDIRECT, any output for DU0: is sent to DU3:, as indicated by the first SHOW DEVICES display. Later, when DU0: is again available, the user issues ASSIGN/REDIRECT again to redirect DU0: to DU0:.

**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****ASS---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 command.

**ASS---Device not redirectable**

**Explanation:** Command attempted to redirect a device that cannot be redirected.

**User Action:** Check command for proper device names and reenter.

**ASS---New device not known to system**

**Explanation:** An ASSIGN/REDIRECT command 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.

**ASS---Old device attached**

**Explanation:** The oldddnn: specified in the command is attached or spooled.

**User Action:** Find the cause of the error and proceed accordingly.

## ASSIGN/REDIRECT (Cont.)

ASS---Old device mounted

**Explanation:** The oldddnn: specified in the command contains a mounted volume.

**User Action:** Wait for device to become available.

ASS---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.

ASS---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.

ASS---TI redirect error

**Explanation:** Command attempted to redirect pseudo device TI:.

**User Action:** None. TI: cannot be redirected.

**DEASSIGN****5.5 DEASSIGN**

DEASSIGN cancels logical name assignments made by the ASSIGN and DEFINE commands.

**Format**

```
DEASSIGN[/qualifier[s]] logical_name
```

**Command Qualifiers**

```
/ALL
/GLOBAL
/GROUP[:g]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:
```

**Parameter**

logical\_name

For systems with extended logical name support, specifies the logical name assignment that you want to delete. This parameter is required with all qualifiers except /ALL.

If you terminate the logical name parameter with one or two colons (:), the command interpreter removes the colons. (Note that the ASSIGN command, also, removes trailing colons, if present, from a logical name before placing the name in the logical name table.) If a colon is present at the end of the actual logical name, you must place quotation marks (") around the logical name parameter for the DEASSIGN command (for example, DEASSIGN "FILE:").

A logical name can contain any ASCII characters. However, if a logical name includes characters other than the 26 letters A through Z, the numbers 0 through 9, an underscore (\_), a colon (:), or a dollar sign (\$), you must place a quotation mark on each side of the logical name. For example, the logical name INFILE does not require quotation marks, but you must specify the logical name C3-PO in the command line as "C3-PO". Generally, any logical name created by the DEFINE command should have the quotation marks placed on either side of the logical name in the DEASSIGN command line.

For systems without extended logical name support, logical\_name specifies the logical device name. 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 by a colon (:).

**DEASSIGN (Cont.)****Command Qualifiers**

Any other qualifier can be used with /ALL. The /TERMINAL qualifier can be used with /LOGIN or /LOCAL.

**/ALL**

Deletes all logical name assignments for a particular table.

If you use /ALL with another qualifier, all assignments of the type specified by the other qualifier are deleted.

Because /ALL deletes all the logical name assignments in a given category, you do not specify the logical name parameter in the command line.

**/GLOBAL****/SYSTEM**

Deletes a global logical name assignment in the system logical name table. This deassignment applies to all tasks running in the system. /SYSTEM and /GLOBAL are synonyms and are privileged qualifiers. You cannot specify /TERMINAL with /GLOBAL.

**/GROUP[:g]**

Deletes a group logical name assignment in the group logical name table. The argument g identifies the User Identification Code (UIC) group number for which the logical name exists. If you do not specify a UIC group number, the default is your own group number, which is taken from your current protection UIC. If you are nonprivileged, the group number is the same as the UIC that is assigned to you when you log in.

**/LOCAL**

Deletes a local logical name assignment in the user logical name table. This is the default.

**/LOGIN**

Deletes a login logical name assignment in the user logical name table.

This is a privileged qualifier.

**/TERMINAL:ttnn:**

Allows you to delete a logical name assignment of another terminal. You cannot specify /TERMINAL with /GLOBAL.

Only privileged terminals can delete assignments made from other terminals.

## DEASSIGN (Cont.)

## Examples

```
$ DEASSIGN TP0: (RET)
```

If the system does not have extended logical name support, this example deletes the local assignment of the logical device name TP0: to a device.

If the system has extended logical name support, this example deletes the local assignment of logical name TP0 to a device. DEASSIGN ignores the terminating colon.

```
$ 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.

```
$ DEAS/LOCAL/ALL/TERMINAL:TT4: (RET)
```

This example deletes all local assignments for terminal TT4:. This command must be issued from a privileged terminal.

```
$ DEASSIGN "XY:" (RET)
```

This example deletes the logical name XY: (created with the DEFINE command). This example is valid only if your system supports extended logical names.

## Notes

DEASSIGN counteracts ASSIGN and DEFINE.

You must type at least the first four characters of the DEASSIGN command.

Login assignments are normally established through ACNT, the Account File Maintenance Program.

You can display assignments with either the SHOW ASSIGNMENTS or SHOW LOGICALS command.

All local assignments are deassigned when you either log out or log

DISMOUNT does not delete logical name assignments.

DEASSIGN/QUEUE is a separate command. See the RSX-11M-PLUS Batch and Queue Operations Manual for more details.

## DEASSIGN (Cont.)

### Error Messages

DEA---Device not terminal

**Explanation:** The /TERMINAL qualifier named a device that is not a terminal.

**User Action:** Retype command after checking for proper syntax.

DEA---Octal group number expected

**Explanation:** You did not specify an octal group number value when you issued the DEFINE/GROUP command.

**User Action:** Check the group number that you want and reenter the command.



**DEFINE****5.6 DEFINE**

The DEFINE command is supported only if extended logical name support was selected during system generation.

The DEFINE command equates a logical name to an explicit ASCII text string (for example, "\$&\* TESTFILE @@6") or to another logical name. All references to the logical name are resolved by the operating system.

Note that DEFINE does not perform validity checks for node names, device names, UFD specifications, or file specifications. If you intend to use a logical name as part of a Files-11 file specification, you should use the ASSIGN command to create the logical name. ASSIGN performs several validity checks to ensure that the logical name will be recognized by the system.

**Format**

```
DEFINE
Logical name?   logical_name
Equivalent name string? equivalence_name

DEFINE[/qualifier[s]] logical_name equivalence_name
```

**Command Qualifiers**

```
/FINAL
/GLOBAL (or /SYSTEM)
/GROUP[:g]
/LOGIN
/LOCAL
/TERMINAL:ttnn:
/TRANSLATION_ATTRIBUTES=TERMINAL
```

**Parameters**

**logical\_name**

The name you selected that you want to give the device or file specification.

A logical name can contain any ASCII characters. However, if a logical name includes characters other than the 26 letters A through Z, the numbers 0 through 9, or the characters underscore (\_), colon (:), or dollar sign (\$), you must place quotation marks (") on each side of the logical name. For example, although the logical name INFILE does not require quotation marks, you must specify the logical name C3PO\* in the command line as "C3PO\*". Unlike the ASSIGN command, the leading and trailing quotation marks are stripped off when the logical assignment occurs.

**DEFINE (Cont.)**

If you specify a colon at the end of the logical name, the DEFINE command saves the colon as part of the logical name. (This is in contrast to the ASSIGN command, which removes the colon before the name in the logical name table.) Note, however, that the system will ignore all logical names terminating with a colon when processing file specifications.

The length of the logical name is limited by the legal length of a command line, which is 79 characters. For example, if the length of DEFINE plus qualifiers plus the equivalence name plus spaces is 24 characters, the logical name cannot be more than 55 characters long.

**equivalence\_name**

An ASCII string (for example, any user-specified text) that you have defined as the substitution for the logical name. The equivalence name can also reference another logical name or a text string.

**Command Qualifiers**

/GLOBAL  
/SYSTEM

Specifies the assignment is to be a system table assignment. The qualifiers /SYSTEM and /GLOBAL are synonyms and are privileged commands. System assignments apply to all tasks running on the system.

/GROUP[:g]

Specifies that the assignment is to a group assignment.

The argument g is the User Identification Code (UIC) group number of the users who share the logical name. If you do not specify a group number, the default is your own group number, which is taken from your current protection UIC. If you are privileged, the group number is the same as the UIC that is assigned to you when you log in.

/LOCAL

Specifies that the assignment is to be a login assignment. This is the default qualifier, so you do not need to specify it.

Commands and tasks initiated from your terminal can access devices and files through the logical names defined for them. Note that no automatic deassignment occurs if you dismount a device after defining a logical name for it.

You can define your set of logical name assignments for your own terminal. Those local logical names exist only for your terminal.

**DEFINE (Cont.)****/LOGIN**

Specifies that the logical definition is to be displayed as a login definition. This is a privileged qualifier.

You can establish login definitions any time during an individual user session. These definitions remain in effect until you log off the system or you specifically delete the definition by using the DEASSIGN/LOGIN command.

Normally, you place login definitions in your login command file, LOGIN.CMD or the system manager places them in the system login file, SYSLOGIN.CMD. Having the logical definitions in either of these login command files saves you having to define those logicals each time you log in. For more information on the SYSLOGIN.CMD file, see the HEL command description in the RSX-11M-PLUS MCR Operations Manual.

**/TERMINAL[:ttnn:]**

Specifies that the requested local assignment be applied to another terminal. Only a privileged user may make assignments to other terminals. Note that the target terminal must be logged in before the assignment can be made.

**/FINAL****/TRANSLATION\_ATTRIBUTES=TERMINAL**

Specifies that the equivalence name string should not be translated iteratively; that is, the logical name translation should terminate with the current equivalence string. See Section 5.2.3.1 for more information on the iterative translation of logical names. The /TRANSLATION\_ATTRIBUTES=TERMINAL qualifier is a synonym for /FINAL, included for VAX/VMS compatibility.

**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 name BIG RIVER @ BEND has a global assignment of RUNS SLOWLY, but a local assignment of RUNS QUICKLY, the operating system interprets BIG RIVER @ BEND to be RUNS QUICKLY for your terminal.

You can counteract the DEFINE command by using the DEASSIGN command or by redefining the logical name.

You can display current assignments with either the SHOW ASSIGNMENTS or the SHOW LOGICALS command.

## DEFINE (Cont.)

### Examples

```
$ DEFINE (RET)
Logical name? "TOM'S TEST" (RET)
Equivalent name string? "BLEW UP" (RET)
$ SHOW LOGICALS (RET)
TOM'S TEST = BLEW UP (local, TT0:)
```

This example defines the logical name TOM'S TEST to the equivalent string BLEW UP. The user issued a SHOW LOGICALS command to verify that the assignment had been made. DEFINE removes the quotation marks from the logical name.

```
$ DEFINE "TOM'S TEST" "BLEW UP" (RET)
```

This example is the same as the previous one.

DEF---Device not terminal

**Explanation:** You did not specify a terminal when you issued the DEFINE/TERMINAL command.

**User Action:** Reenter the command line specifying a terminal.

DEF---Function requires logical name support.

**Explanation:** Your system does not support extended logical name support.

**User Action:** None.

DEF---Octal group number expected

**Explanation:** You did not specify an octal group number value when you issued the DEFINE/GROUP command.

**User Action:** Check the group number that you want and reenter the command.

DEF---Terminal not logged in

**Explanation:** The terminal you attempted to make a logical definition for is not logged in.

**User Action:** Log in the terminal you want to make the definition for and reenter the command line.

**ALLOCATE****5.7 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 Qualifiers**

```
/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 want 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.

## ALLOCATE (Cont.)

### Examples

```
$ ALLOCATE DU RET
ALL -- DU0: Now allocated
```

This example allocates the first available DU drive (RD51 or RX02). The message gives the name of the device allocated. This is an example of a generic allocation.

```
$ ALLOCATE RET
Device? DU1: RET
```

This example allocates the specified device. The allocation is confirmed by the return of the prompt.

```
$ ALL DU1: RET
ALL -- Device allocated to other user
```

This example shows an allocation that failed because the device named was not available.

```
$ ALL DL1: TP0: RET
ALL -- DL1: Now allocated
```

This example allocates the specified device and assigns it the logical name TP0:.

```
$ ALLOCATE/TYPE:RD51 DU RET
ALL -- DU1: Now allocated
```

This example shows the use of the /TYPE qualifier. The user may want to allocate the first available RD51 but to ignore any other devices using the DU driver. This is also a generic allocation.

### Notes

When you allocate a device, no other user can gain access to the device.

REALLOCATE counteracts ALLOCATE.

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.

You cannot include ALLOCATE in RSX-11M-PLUS batch jobs. See the RSX-11M/M-PLUS Batch and Queue Operations Manual for information on private devices in batch jobs.

## ALLOCATE (Cont.)

### Error Messages

ALL---Device allocated to other user

**Explanation:** Command attempted to allocate device already allocated by another user.

**User Action:** Wait until the user has deallocated the device before you allocate 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 until task is finished before allocating the device.

ALL---Pseudo device error

**Explanation:** Command attempted to allocate a pseudo device. Pseudo devices cannot be allocated.

**User Action:** Retype command after checking for proper device name.

ALL---Public device

**Explanation:** Command attempted to allocate a public device.

**User Action:** None. Public devices cannot be allocated.

ALL---User logged on terminal

**Explanation:** Command attempted to allocate a terminal logged in by another user.

**User Action:** None. Only terminals that are not logged in can be allocated.

# DEALLOCATE

## 5.8 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
/TERMINAL:ttnn:
```

### Parameter

ddnn:

Specifies the device to be deallocated.

### Command Qualifiers

/ALL

Frees all devices allocated from the terminal being used. If you use this qualifier, you cannot specify a device name.

/DEVICE

This qualifier is non-operational.

/TERMINAL:ttnn:

Enables a privileged user to deallocate devices allocated from another terminal.

### Example

```
$ DEALLOCATE (RET)
Device? DU0: (RET)
```

This example deallocates the device DU0: 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,



## DEALLOCATE (Cont.)

### 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.

DEA---Device mounted

**Explanation:** Command attempted to deallocate a mounted device.

**User Action:** You cannot deallocate a mounted device. Dismount the device.

# MOUNT

## 5.9 MOUNT

MOUNT declares a volume to be logically known to the system, and sets the volume online 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

```
MOUNT[/qualifier[s]] ddnn: volumelabel
```

#### For Magnetic Tapes

```
MOUNT[/qualifier[s]] ddnn:[,ddnn:...] fileset-ID
```

#### Command Qualifiers for Both Disks and Tapes

```
/DEFAULT:arg
      SAVE
      [NO]UNLOAD
/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:[g,m]
/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
```

**MOUNT (Cont.)**

```

/DENSITY:arg
    800
    1600
/[NO]HDR3
/[NO]LABEL
/OVERRIDE:(arg[,s])
    ACCESSIBILITY
    EXPIRATION_DATE
    IDENTIFICATION
    SET_IDENTIFICATION
/RECORD_SIZE:n
/TRANSLATE:arg
    EBCDIC
    NONE
    UT1
    UT2
    UT3
/VOLUME_IDENTIFICATION:(volume-ID[,volume-ID[,s]])

```

**Parameters**

ddnn:

Specifies the 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 5.9.1.

**NOTE**

Disks and magnetic tapes differ in the way labeling is handled. See the following discussions of volumelabel and fileset-ID. For more information on magnetic tape Volume IDs and File Set IDs, see Section 5.9.1.

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 want to mount. Disk and random-addressable volume labels can be as many as 12 characters.

For disks, volume labels (or identifiers) can include any alphanumeric character without restriction. For ANSI-standard magnetic tapes, volume labels 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 ("), the quotation mark 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 6 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.

### Command Qualifiers for Both Disks and Tapes

/DEFAULT:arg  
SAVE  
[NO]UNLOAD

Specifies the default that will apply when the volume is dismounted. SAVE is a privileged argument and specifies that the medium is not unloaded and can be accessed for reads and writes by privileged tasks. UNLOAD specifies that the medium is to be unloaded after the dismount. This is the default for Files-11 volumes. NOUNLOAD specifies that the medium is not to be unloaded. This is the default for foreign volumes.

Only privileged users can specify /DEFAULT = UNLOAD for a multiunit device.

This qualifier applies to DB:, DM:, DU:, and magnetic tape devices only.

Remember, these values can also be set when you issue the DISMOUNT command. (See Section 5.10.)

/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 Chapter 4, for more information on file protection codes and the relationship between file protection and volume protection. See also the /PROTECTION qualifier.

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.

**MOUNT (Cont.)****/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. On RSX-11M-PLUS, this qualifier is required for mounting foreign volumes. 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 apply only to magnetic tape.

**/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 (the argument acpname) 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:

1. ddnnF1
2. ddAACP
3. F11ACP

Thus, for an RX50 device, which uses the device mnemonic DU:, MOUNT looks first for an ACP named DUnnF1, where nn is the device number. If no ACP of that name exists, MOUNT looks for an ACP named DUAACP and, finally, for F11ACP.

## MOUNT (Cont.)

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:

1. ddnnF1
2. ddAACP
3. F11ACP

Thus, for a TU58 DECTape, which uses the device name DD:, MOUNT looks first for an ACP named DDnnF1, where nn is the device number. If no ACP of the appropriate name exists, MOUNT looks for an ACP named ddAACP. Failing that, MOUNT looks for 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 TSV05 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.

You only have to specify an ACP in special cases. For example, 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. In this case, you will also have to include the /FOREIGN qualifier. You may also need to use the /PARAMETERS qualifier to enter parameters used by that ACP.

The UNIQUE argument specifies that a dedicated ACP be used for the volume being mounted. MOUNT creates a copy of the F11ACP if the device is a disk or DECTape, or a copy of the MTAACP if the device is a magnetic tape. This unique ACP will be given a name of the form ddnnF1. When the volume is dismounted and the file processor exits, the ACP is automatically removed.

If the ACP (F11ACP or MTAACP) is installed in secondary pool, the /PROCESSOR:UNIQUE qualifier is the default.

You can control the resources dedicated to file processing by choosing appropriate names for installing ACPs. For example, if you want DL0: to have a dedicated ACP, and DL1: and DL2: to use a shared ACP, install the F11ACP as DL00F1.

If you mount a volume /FOREIGN but do not specify an ACP, no ACP can access the volume. However, tasks such as INITIALIZE or user tasks with nonfile-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 Chapter 4 for more information on volume protection.

**MOUNT (Cont.)**

**/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.

**/[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.

If you specify **/NOSHAREABLE** explicitly when mounting a volume on a device, the device is allocated and set nonpublic.

If the device is already allocated, the default is **/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.

**MOUNT (Cont.)**

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.

The information displayed by /SHOW is similar for magnetic tapes. 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.

**/[NO]WAIT**

Specifies whether you require operator assistance in performing the mount. This qualifier is used mainly for batch jobs, indirect command files, or terminals distant from the machine room. The default is /WAIT for mounts in batch jobs and indirect command files, and /NOWAIT for interactive mounts.

If the mount is included in a batch job or indirect command file, or if you specify /WAIT in an interactive mount, a message concerning the mount is sent to the operator's console; 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.)

For more information on mounts from batch jobs, see the discussion of the MOUNT command in the RSX-11M/M-PLUS Batch and Queue Operations Manual.

**/[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.



**MOUNT (Cont.)****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 range from 1 through 127. The default is the value specified when the volume was initialized.

**/OWNER:[g,m]**

Specifies the owner (UIC) 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 the file and volume protection.

**/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 only be read.

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 can be the numbers 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, using the USER and INDEX arguments, respectively. If you specify separate values, the USER argument sets the value for the user default as well as the volume default. The system creates the index file window using the value specified for the INDEX argument.

## MOUNT (Cont.)

The default for /WINDOWS is the value specified in the Volume Home Block. This value is established when the volume is initialized. If you want 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 range 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  
                   FORTRAN  
                   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.

/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 when it is known that the target system cannot handle a tape with a HDR3 label (the default). This qualifier has no other effect or application.

/[NO]LABEL

Specifies whether the tape volume is ANSI or unlabeled. An ANSI tape is specified by the /LABEL qualifier (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

**MOUNT (Cont.)**

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 tries to write a short block (less than 18(decimal) bytes) because the data did not fill the block, RMS writes an even-aligned (even-byte boundary) block of 18(decimal) bytes.
- In the case of blocks less than 18(decimal) bytes, RMS 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 ignores all circumflex records that have a FIXED format.

```
/OVERRIDE:(arg[,s])
    ACCESSIBILITY
    EXPIRATION DATE
    IDENTIFICATION
    SET_IDENTIFICATION
```

Specifies that you want 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 want to specify only one argument. If you want to specify more than one argument, the arguments must be enclosed in parentheses and separated by commas. The arguments are as follows:

- ACCESSIBILITY---Specifies that you want to override the protection on ANSI magnetic tapes.
- EXPIRATION\_DATE---Specifies that you want to override the expiration date on ANSI magnetic tapes.
- IDENTIFICATION---Specifies that you want 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.
- SET\_IDENTIFICATION---Specifies that you want 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 range from 1 through the value for blocksize on the volume.

## MOUNT (Cont.)

```
/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 specifies user-supplied translation tables that have been previously linked into the MTAACP. See the RSX-11M/M-PLUS and Micro/R SX I/O Operations Reference Manual for further information on character translation.

```
/VOLUME_IDENTIFICATION: (volumeID[,volumeID[,s]])
```

Specifies Volume Identifiers. If you want 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 5.9.1 for more information, including examples of this qualifier in use.

### Examples

```
$ MOUNT (RET)
Device? DL1: (RET)
Volume ID? HOTROD (RET)
```

This example mounts the volume labeled HOTROD on device DL1:. If you are doing nothing more than mounting a volume to read from or write to, 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)
```

This example mounts a 3-volume ANSI magnetic tape set on devices MM0:, MM1:, and MM2:. The volume labels are W\*MB\*T, FOOBAR, and FOO. The first label is included within quotation marks to allow the specification of the asterisk (\*). Note that the comma between "W\*MB\*T" and FOOBAR is outside the quotation marks.

## MOUNT (Cont.)

```
$ 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; therefore, 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- (RET)
->/TRANSL:EBCDIC MS1: (RET)
```

This example mounts an unlabeled magnetic tape on device MS1:, specifying a block size of 2048 characters and a record size of 64 characters. The volume includes FORTRAN carriage control, so that is specified as well. 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 then translated back to EBCDIC for writing to the tape volume.

```
$ MOUNT/WAIT DL1: WHIZZER (RET)
```

In this example, the user wants an operator to place the medium containing the proper volume in place on DL1:. The command may appear, in a batch job, 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 DL1:

Type

UNS (DCL START) when ready

or

RES (DCL CONTINUE) to reject request.
```

The operator performs the requested operation and 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. The batch processor attempts to continue the job. The same is true if the command appeared in an indirect command file.

**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. However, mounting enables 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 for that device.

## MOUNT (Cont.)

If you attempt to mount a volume on a device where the disk is not properly spun up, MOUNT waits 30 seconds and tries 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.

DISMOUNT counteracts MOUNT. See Section 5.10 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 5.11.

### 5.9.1 Mounting Magnetic Tapes and Tape Sets

Mounting of magnetic tape volumes is handled by a special ACP, the MTAACP.

The MOUNT command can mount only 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 can mount magnetic tapes by the set, then, 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.

## MOUNT (Cont.)

If you want 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 MS0: FOO (RET)
```

mounts a tape set (consisting of a single physical reel) whose File Set ID is F00 and whose Volume ID is F00. The reel F00 must be loaded on device MS0: at the time of the mount.

The following command:

```
$ MOUNT MS0: DRAG/VOLUME IDENTIFICATION: (FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is DRAG) and that consists of three physical reels with Volume IDs of F00, BAR, and MORE. The reel with the Volume ID F00 must be loaded on device MS0:, 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 F00) does not take place until the first record is written to the tape.

The following command:

```
$ MOUNT/OVERRIDE:SET_IDENTIFICATIONMS0: (FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is immaterial) that consists of three physical reels with Volume IDs of F00, BAR, and MORE. The reel F00 must be loaded on device MS0:, but the other two reels are not loaded at the time of the mount.

The following command:

```
$ MOUNT MS0: FOO/VOLUME_ID: (A312,A387,B231) (RET)
```

mounts a tape set (whose File Set ID is F00) that consists of three physical reels with Volume IDs A312, A387, and B231. The reel A312 must be loaded on device MS0:, but the other two reels are not loaded at the time of the mount.

The following command:

```
$ MOUNT (MS0:,MS1:,MS2:) (FOO,BAR,MORE) (RET)
```

mounts a tape set (whose File Set ID is F00) that consists of three physical reels with Volume IDs F00, BAR, and MORE. At the time of the mount, the reel F00 must be loaded on device MS0:, the reel BAR must be loaded on device MS1:, and the reel MORE must be loaded on device MS2:.

## MOUNT (Cont.)

### NOTE

There is no theoretical limit to the number of physical tape reels in a set, but your DCL MOUNT command must translate to a MCR MOU command of 80 characters or fewer. If your DCL MOUNT command line is more than 80 characters, you should use the MCR MOU command. Remember, you can use the MCR command to enter MCR commands from a DCL terminal.

### Error Messages

MOU---Failed to start Ancillary Control Processor

**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.

MOU---Device specified twice

**Explanation:** The command specified the same device twice.

**User Action:** Retype command after checking for 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.



## MOUNT (Cont.)

### MOU---Home block I/O error

**Explanation:** An I/O error was detected while 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 for 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.

### 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 DEVICES 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 that you have the correct magnetic medium, or that the volume has not been initialized. See your system manager.

## MOUNT (Cont.)

MOU---Task not ACP

**Explanation:** The task named with the /PROCESSOR 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 for 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 for 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 for 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 for proper syntax.

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.

## MOUNT (Cont.)

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

### 5.10 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 5.9. 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. The /SYSTEM qualifier is a synonym for /PUBLIC, included for compatibility with VAX/VMS.

## DISMOUNT (Cont.)

/SAVE

Specifies that the disk is to remain spinning in the drive and can be accessed for read or write by privileged tasks.

This is a privileged qualifier that applies to DB:, DM:, and DU: devices only.

The most common use of /SAVE is when saving or backing up the system disk. If you simply want to leave the disk spinning, use the /NOUNLOAD qualifier. Remember, these values can also be set with the /DEFAULT qualifier to MOUNT. See Section 5.9.

/TERMINAL:ttnn:

Allows a privileged user to dismount a volume mounted from another terminal.

/[NO]UNLOAD

Specifies that the disk is to spin down after the dismount. This is the default for Files-11 volumes. For magnetic tapes, this qualifier allows the tape to be completely rewound.

You can spin down the removable unit of a multiunit device only if you are a privileged user. If you are a nonprivileged user and you try to dismount the unit or specify /UNLOAD in the command line, you will get a warning message that the volume cannot be spun down.

The /NOUNLOAD qualifier specifies that the disk is to remain spinning after the dismount. This is the default for foreign mounted volumes. The /NOUNLOAD qualifier is useful when the volume is to be immediately remounted. For magnetic tapes, this qualifier rewinds the tape to the first file position. Remember, these values can also be set with the /DEFAULT qualifier to MOUNT.

The /UNLOAD and /NOUNLOAD qualifiers apply to DB:, DM:, and DU: disk drives and magnetic tape drives only.

### Examples

```
$ DISMOUNT (RET)
Device? DL0: (RET)
```

```
DIS -- TT3: Dismounted from DL0: *** Final Dismount Initiated ***
```

This example dismounts a volume on device DL0:. No other user had the volume mounted when the DISMOUNT command was issued.

```
$ DISMOUNT DU1: HOTROD (RET)
```

```
DIS -- TT4: Dismounted from DU1:
```

This example dismounts a volume on device DU1:. The user specified the label HOTROD to be sure the correct volume was mounted on the device. The volume was mounted shareable and because another user also has it mounted, the informational message does not include the notification of final dismount.

## DISMOUNT (Cont.)

```
$ LOGOUT (RET)
DMO -- TT1: Dismounted from DU0: *** Final dismount ***
Have a good morning
22-APR-85 TT1: 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 out. LOGOUT dismounted the volume. If the device was allocated, LOGOUT also deallocated it. This is the equivalent of DISMOUNT/ALL.

```
$ DISMOUNT/ALL (RET)
DMO -- TT1: Dismounted from DU0:
DMO -- TT1: Dismounted from DU1: *** Final dismount ***
DMO -- TT1: Dismounted from MS1: *** Final dismount ***
```

This example dismounts all volumes mounted from the terminal at which the DISMOUNT was issued. See next example.

```
$ DISMOUNT/ALL (RET)
DMO -- TT1: Dismounted from DU0:
DMO -- TT1: Dismounted from DU1: *** Final dismount ***
DMO -- TT1: Dismounted from MS1: *** Final dismount ***
$ MOUNT DY1: 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 DU1:. Because this volume was necessary for a running task, the user immediately issued a new MOUNT command using the volume label CRUCIAL. As 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 DL0: (RET)
DMO -- TT2: Dismounted from DL0:
DMO -- TT1: Dismounted from DL0:
DMO -- TT3: Dismounted from DL0:
DMO -- TT5: Dismounted from DL0: ***Final dismount ***
```

In this example, a privileged user dismounted all users who had mounted the volume on DL0:.

### 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 AC P 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 that are 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:** None.

DMO---Warning---Volume will not spin down <ddnn:>

**Explanation:** An attempt was made by a nonprivileged user to spin down the removable unit of a multiunit device, either implicitly with the DISMOUNT command or explicitly by specifying /UNLOAD in the command line. Only privileged users can spin down multiunit devices. (For nonprivileged users, the volume is logically dismounted but not spun down.)

**User Action:** None.

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.

## DISMOUNT (Cont.)

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****5.11 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
```

## INITIALIZE (Cont.)

### Parameters

ddnn:

Specifies the name of the device on which the magnetic medium to contain the volume has been placed.

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. See Section 5.12 for use of the volume label with the BACKUP command.

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 5.9, 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

Specifies the number of directories that will be accessed simultaneously. FllACP 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 (MFD), nor do directory operations require that the directory file header be read. In general, you can significantly speed up directory operations 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 by the ANALYZE/MEDIA command and automatically determines the bad-block information for the volume. This is the default.

**INITIALIZE (Cont.)**

(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.

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.

This value can be altered on an existing volume with the INITIALIZE/UPDATE command. (See Section 5.9.1.)

Here are the acceptable density values for DIGITAL devices available on RSX-11M-PLUS 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

**INITIALIZE (Cont.)**

Mnemonic	Device Densities	
MM: (magnetic tape)	TU77	800 (default) 1600
DY: (diskette)	RX02	LOW (or single) HIGH (or double) (default)

**/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.

**/FILE\_PROTECTION: (code)**

Specifies the default protection for all files on the volume being initialized. See Chapter 4, 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.

**/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	File Headers	
	Device Allocated Initially (n)	
DB: RP04/RP05		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/RM03		4049
DR: RM05		25593
DR: RM80		7461
DR: RP07		51699
DS: RS03/RS04		16

## INITIALIZE (Cont.)

Mnemonic	File Headers Device Allocated Initially (n)
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 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 1-, 2-, or 3-header index file.

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:**

$$\text{headers} = \text{maximum}/2$$

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:**

$$\text{headers} = \text{maximum}/2 \text{ or } 25593(\text{decimal}), \text{ whichever is larger}$$

**three-header index file:**

$$\text{headers} = \text{maximum}/2 \text{ or } 51699(\text{decimal}), \text{ whichever is larger}$$

See the description of the /MAXIMUM\_FILES qualifier for more information on multiheader index files.

**INITIALIZE (Cont.)**

```

/INDEX:arg
    BEGINNING
    MIDDLE
    END
    n

```

Specifies the location of the index 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.

```

/LABEL: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:

$$\text{default-MAXIMUM\_FILES} = ((X - ((X + 4095.) / 4096.) + 9.) * 127.) / 258.)$$

## INITIALIZE (Cont.)

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-PLUS V2.1. INITIALIZE issues a warning message about incompatibility when it creates a structure level 402 volume.

The following table includes the default number of files and the maximum number of files for each device.

Mnemonic	Device	Default n	Maximum n
DB:	RP04/RP05	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/RM03	8099	25600
DR:	RM02/RM03	8099	64798
DR:	RM05	30781	65500
DR:	RM80	14923	65500
DR:	RP07	62007	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, INDEX.SYS, and 000000.DIR) required for Files-11 structure. See the RSX-11M/M-PLUS and Micro/RSX 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.

**INITIALIZE (Cont.)****/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 Chapter 4, 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.

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 seconds is also dropped. For example:

20MAR8510453

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. 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 want to see all the values with which the volume is being initialized. /NOSHOW is the default.



## INITIALIZE (Cont.)

These values are the same as the following DCL qualifiers:

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

/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.

## Examples

```
$ INITIALIZE (RET)
Device? DU0: (RET)
Volume ID? HONOLULU (RET)
```

This example initializes a volume with the volume label HONOLULU on device DU0:. See the next example.

```
$ ALLOCATE DU0: (RET)
$ MOUNT/FOREIGN DU0: (RET)
$ INITIAIZE DU0: HONOLULU (RET)
$ DISMOUNT DU0: (RET)
$ MOUNT DU0: 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 has already been written on the medium.)

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.

## INITIALIZE (Cont.)

```
$ INIT/INDEX:END/PROTECTION:(SYSTEM:RWED,OWNER:,GROUP:,- RET)
-> WORLD:) DU1: MIRACLE RET
```

This example initializes a volume named MIRACLE on a medium mounted on device DU1:. 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

All users must allocate the device on which the volume is to be initialized.

On RSX-11M-PLUS, you must use MOUNT/FOREIGN before INITIALIZE.

### 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.

## INITIALIZE (Cont.)

### 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 for proper syntax.

### 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.

## INITIALIZE (Cont.)

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 ANALYZE/MEDIA command 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.

INI---Not file structured device

**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:** None. 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 [000000]BITMAP.SYS.

**User Action:** Use the ANALYZE/MEDIA command on the disk again to find bad blocks. Then retry the operation.

**INITIALIZE (Cont.)****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 the diskette in RX02 (DYnn:) drive be initialized at low density, but the diskette was formatted at high density.

**User Action:** Reformat the diskette, or change it.

**INI---Unit is not at low density**

**Explanation:** Command specified that the diskette in RX02 (DYnn:) drive be initialized at high density, but the diskette was formatted at low density.

**User Action:** Reformat the diskette, or change it.

**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 /PROCESSOR qualifier.

**User Action:** Remount the volume without the /PROCESSOR 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

### 5.11.1 INITIALIZE/UPDATE

INITIALIZE creates a Volume Home Block that includes a number of values used by the ACP when you mount the volume. INITIALIZE/UPDATE alters 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
      800
      1600
/EXTENSION:n
/FILE PROTECTION:code
/LABEL:newvolumelabel
/MAXIMUM FILES:n
/OWNER:[g,m]
/PROFESSIONAL
/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 the 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.

```
/DENSITY:arg
      800
      1600
```

Alters the /DENSITY value established when the volume was initialized. See the description of the /DENSITY qualifier under INITIALIZE (Section 5.11) for more information.

**INITIALIZE/UPDATE (Cont.)****/EXTENSION:n**

Specifies the number of blocks by which a file will be extended if it has exhausted its space allocation. The value for n can range from 1 through 127. The default is set by your system manager.

**/FILE\_PROTECTION:code**

Specifies the protection for new files created on the volume. See Chapter 4, for more information on file protection.

**/LABEL:new volumelabel**

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 5.9) for more information about volume labels and the characters that 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 INITIALIZE 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 5.1.3 for more information.

**/PROFESSIONAL**

Applies different defaults to a volume when the volume is initialized. The qualifier assumes the volume is an RX50.

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 labeled 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, not including hyphens and colons. The second digit for seconds is also dropped. For example:

20MAR8510453

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 and the default directory for the files is [200,200] instead of [1,1]. These defaults cannot be changed.

## INITIALIZE/UPDATE (Cont.)

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 volume protection for the volume. See Chapter 4, 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 for DU1 \*\*

```
Label      = MUSTARD
/CHA       = [ ]
/Ext       = 5.
/Fpro      = [RWED,RWED,RWED,R]
/Lru       = 3.
/Mxf       = 48.
/Pro       = [RWED,RWED,RWED,RWED]
/Uic       = [1,1]
/Win       = 7.
```

Maximum possible files = 4096

The Homeblock revision count is 0

Volume created on 23-APR-85 at 14:14:42

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.

/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.

### Notes

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.



## INITIALIZE/UPDATE (Cont.)

### 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 cannot 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 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:** INITIALIZE/UPDATE 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 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

### 5.12 BACKUP

The BACKUP command is a DCL command which backs up from and restores files to Files-11 volumes. You can use the BACKUP command to transfer some or all of the contents of a disk volume to one or more disks, diskettes, or tapes. This is called backing up your data.

If the data on original disk is corrupted or lost, you can use the BACKUP command with the backup media to retrieve the information. This is called restoring your data.

To perform backup and restore operations, the BACKUP command uses the Backup and Restore Utility (BRU). You might see error messages from either BACKUP or the BRU utility in the course of using the BACKUP command. Section 5.12.4 lists possible error messages for both BACKUP and BRU.

This section describes the BACKUP command format and the available qualifiers. The information in this section is provided for users who need to develop their own backup procedures to fit particular applications.

#### 5.12.1 On-Line BACKUP Diskette, Disk, and Tape Device Information

BACKUP uses disks, diskettes, and tapes as media for its backup and restore operations. Input disks and diskettes that are being backed up must be in Files-11 format. However, when creating backup sets for tapes and multivolume disks and diskettes, BACKUP uses its own format.

BACKUP backs up disks and diskettes that are mounted as Files-11 volumes or as foreign volumes.

BACKUP does not use the Files-11 file system on input disks or diskettes. However, when you are using a mounted volume for a backup operation, BACKUP checks the read access privileges of directories and files against the UIC under which BACKUP is running. To back up from a mounted disk volume that is in Files-11 format, you must specify the /MOUNTED qualifier. For disks or diskettes mounted foreign, no qualifier is necessary.

BACKUP also restores to Files-11 or mounted foreign disk or diskette volumes. Specify the /INITIALIZE qualifier to restore to a volume mounted foreign. This qualifier initializes the volume to a Files-11 format. To restore to a mounted Files-11 volume, specify the /NOINITIALIZE qualifier to indicate to BACKUP that the disk or diskette is mounted and in Files-11 format.

BACKUP backs up to and restores from tape volumes. The tapes must be mounted foreign, but no qualifier is necessary.

**BACKUP (Cont.)****5.12.2 The Backup Command Line****General Format:**

```
BACKUP[/qualifiers]
From?   ddn:
To?     ddn:
```

```
BACKUP[/qualifiers] ddn:[filespecification[s]] ddn:
```

Each time you issue the BACKUP command, you must specify the input device and the output device in the form ddn:.

The input device contains the data you want to transfer. The output device is the device to which data is being transferred.

Note that you can specify a particular tape, diskette, or disk drive only once, but you may specify as many as eight on each command, provided you have those drives.

For tapes and diskettes, if you are using the /IMAGE qualifier, you may specify more than one input or output device (separated by commas) using one of the following formats:

```
BACKUP dd0: dd1:
```

```
BACKUP/IMA:SAV dd0:,dd1: dd2:
```

Your file specification is used to select particular files or categories of files to back up or restore. A file specification takes the following form:

```
[directory]filename.typ;ver
```

You may specify as many as 16(decimal) file specifications per command. Files can be backed up or restored selectively by directory, 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 field that is not specified is taken as a wildcard. For example, if you do not specify a 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.

Refer to Section 5.12.2.1 for a list of the qualifiers available with the BACKUP command.

**5.12.2.1 BACKUP Command Qualifiers**

BACKUP provides six groups of command qualifiers. Unless otherwise noted, the defaults for these qualifiers are the same as the equivalent attribute of the input volume.

## BACKUP (Cont.)

The six groups of qualifiers are as follows:

### Group 1: Selective Backup and Restore

#### Command Qualifiers

```

/CREATED:BEFORE:(dd-mmm-yy nn:mm)
                AFTER:(dd-mmm-yy nn:mm)
/EXCLUDE
/IMAGE:arg
                SAVE
                RESTORE
/NEW_VERSION
/MODIFIED:arg
                BEFORE:(dd-mmm-yy nn:mm)
                AFTER:(dd-mmm-yy nn:mm)
/[NO]REPLACE

```

### Group 2: Initialization

#### Command Qualifiers

```

/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

#### Command Qualifiers

```

/APPEND
/DENSITY:arg
                800
                1600
/ERROR_LIMIT:n
/LENGTH:n
/REWIND
/LABEL:TAPE:fileset-ID

```

**BACKUP (Cont.)****Group 4: Verification****Command Qualifiers**

/COMPARE  
/VERIFY

**Group 5: Display****Command Qualifiers**

/[NO]LOG  
/LIST

**Group 6: Disk and Diskette Processing****Command Qualifiers**

/APPEND  
/DIRECTORY  
/LABEL:INPUT:volumelabel  
/NOINITIALIZE  
/MOUNTED  
/[NO]PRESERVE  
/LABEL:OUTPUT:volumelabel

Sections 5.12.2.2 to 5.12.2.7 describe the various groups of qualifiers.

**5.12.2.2 Selective Backup and Restore Qualifiers**

This section describes the qualifiers that you may use to perform selective backup and restore operations.

/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 may specify a date, a time, or both. If you do not specify a date and time, the default is your system's current time and date. Here are some examples:

/BEFORE:31-MAY-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 want to enter both a date and a time, you must use the parentheses. These formats are used wherever you specify a date, 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]*.* DU0: RET
```

This command transfers all files except those in the directory [306,6].

### /IMAGE:arg SAVE RESTORE

Specifies that you want to use more than one diskette or disk during a backup or restore operation.

You must use the /IMAGE:SAVE qualifier when you create a backup set that spans more than one disk or diskette. For example, if you are backing up a large disk to several small disks, you must use this qualifier.

You must use the /IMAGE:RESTORE qualifier when you are restoring a backup set that spans multiple disks or diskettes. For example, if you are restoring a backup set to a large disk, and your backup set spans six smaller disks, you must use this qualifier when you restore the backup set to the large 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 during restore operations or 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, a time, or both.

The required date/time argument can be in several formats. You must specify a date, a time, or both. If you do not specify a date or time, the default is your system's current date and time. Here are some examples

**BACKUP (Cont.)**

```

/BEFORE:31-MAY-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 want to enter both a date and a time, you must use the parentheses.

**/[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 the file 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.

**5.12.2.3 Initialization Qualifiers**

This section describes the qualifiers you must use during initialization.

**/INITIALIZE**

Specifies that you want to initialize the output disk before proceeding with the data transfer. You must specify this qualifier if you want 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 or not you want to initialize before proceeding. The default is **/NOINITIALIZE**.

See Section 5.11 for a description of the DCL command **INITIALIZE** and its qualifiers.

```

/BAD_BLOCKS:arg
    AUTOMATIC
    MANUAL
    OVERRIDE

```

Enters the locations of bad blocks on the volume being initialized. See Section 5.3 for more information on finding and marking bad blocks on diskettes using the DCL command **ANALYZE/MEDIA**.

**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.

**BACKUP (Cont.)**

MANUAL specifies that bad-block information for the volume is to be entered manually.

OVERRIDE applies only to last-track devices. It makes the diskette appear not to have bad-block information written in the last track even if the 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 in Section 5.9 for more information.

**/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 5.9. The description also includes the default value of the argument 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. The arguments are as follows:

- BEGINNING---Specifies the beginning of the volume.
- MIDDLE---Specifies the middle of the volume. This is the default for disks and diskettes.
- END---Specifies the end of the volume.
- n---Specifies the logical block n.



**BACKUP (Cont.)**

The default file 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 Chapter 4, for more information on file protection.

The default system protection code is as follows:

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

The default 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 5.9 for more information.

**/SAVE\_SET:name**

Allows you to create a different backup set name for each backup set that you place 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 by BACKUP.

If you do not use this qualifier, the backup set name for a tape or an unmounted disk or 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 several 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 the /REWIND qualifier is specified.

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.

## BACKUP (Cont.)

Note that all backup sets must begin on the first tape, diskette, or disk 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 on the first diskette or disk. The sixth backup set must also begin on the first reel of tape, or the first diskette or disk, but it may span more than one tape, diskette, or disk.

### 5.12.2.4 Tape Control

The following qualifiers allow you to control the way the operating system manages tape data transfer operations.

#### /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 the output to be written where the tape is currently positioned. If you want to start writing at the logical end-of-tape, you must specify /APPEND/REWIND, which rewinds the tape and then spaces forward to the end of the volume.

You can also append new data to a disk or diskette volume. See Section 5.12.2.7 which describes disk and diskette processing.

#### /DENSITY:arg 800 1600

Specifies the density in bits per inch (bpi) for the medium loaded on the device. If you are adding your own tape drives that allow other densities, you must use this qualifier. See Section 5.9 for a list of acceptable densities for DIGITAL devices.

#### /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 volume 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.

**BACKUP (Cont.)****/REWIND**

Rewinds the first tape of a tape backup set before executing the command line.

See the /APPEND qualifier in this section for more information on REWIND.

**/LABEL:TAPE:fileset-ID**

Specifies a 6-character ANSI File Set ID. On output operations, the 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.

**5.12.2.5 Verification Qualifiers**

The following qualifiers allow you to verify the accuracy of data transfers.

**/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 ensure 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.

**5.12.2.6 Display Qualifiers**

The following qualifiers display information about the data transferred.

**/[NO]LOG**

Displays at your terminal the directory 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 file on the specified diskette volume. In a multivolume backup set, the directory is on the first tape, diskette, or disk of the set. See the examples in Section 5.12.3.

## BACKUP (Cont.)

### 5.12.2.7 Disk Processing Qualifiers

The following qualifiers allow you to control the way the operating system manages disk or diskette 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 Section 5.12.2.4 which describes tape control.

You cannot use the /APPEND qualifier during a backup operation to a mounted disk or diskette.

#### /DIRECTORY

Directs BACKUP to create directories (if they do not already exist) on a mounted output volume, and then copy into those directories the files from the same directory on the input volume. If you do not specify /DIRECTORY, BACKUP does not copy the files. /DIRECTORY is used only with the /NOINITIALIZE qualifier.

#### /LABEL:INPUT:volumelabel

Specifies the volume label of the input diskette. This is not a required parameter, but if you specify it, BACKUP checks the volume label.

#### /NOINITIALIZE

Specifies that you do not want to initialize the output diskette, 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 diskette 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 operation using disks or diskettes. The default is /PRESERVE.

#### /LABEL:OUTPUT:volumelabel

Specifies a volume label for the output diskette. The default is the label of the input volume.

## BACKUP (Cont.)

## 5.12.3 Examples

The following are examples of various BACKUP operations.

## Examples

```
$ BACKUP/LABEL:INPUT:RANDOM/SAVE_SET:19APR85 (RET)
From? DU0: (RET)
To: DU1: (RET)
```

```
BAC - Starting disk 1 on DU1:
BAC - End of disk 1 on DU1:
BAC - Completed
```

This example transfers all files on the diskette labeled RANDOM to the diskette on DU2:. The backup set name 19APR85 identifies the data written on the output disk.

```
$ BACKUP/LIST MS0: (RET)

VOL1 18JUL84 INPUT1 18-APR-85
VOL1 BACKUP2 INPUT2 19-APR-85
VOL1 19JUL84 RANDOM 19-APR-85
```

This example shows a directory of tape volume. It displays the tape volume number (VOL1), the names of three volumes, and the names of the three backup sets contained on the tape.

```
$ BACKUP/SAVE_SET:19APR85/LIST DU1: (RET)
VOL1 19APR85 RANDOM 19-APR-85 23 37
[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 DU1:
```

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 previous example).

```
$ BACKUP/REWIND/SAVE_SET/:18APR85/NOINITIALIZE MS0: DU0: (RET)
```

This example rewinds the tape on drive MS0:, then locates the backup set with the name 18APR85 (there may be more than one backup set on a tape) and transfers all files in the backup set to DU0:. The /NOINITIALIZE qualifier instructs BACKUP not to initialize DU0:.

**BACKUP (Cont.)**

\$ **BACKUP/REVISED:AFTER:(14-MAY-85 17:00)/VERIFY DU0: MS0:** **(RET)**

This example backs up all files on the fixed disk that were revised after 5:00 P.M. on May 14, 1985. After all the files have been copied onto the tape, BACKUP verifies the tape. If files on the tape do not verify, BACKUP returns an error message.

\$**BACKUP/INI/IMAGE:SAV/VER/MOU DU0: DU1:** **(RET)**

BAC - Mount disk 1 on DU1: . Press "RETURN" when done **(RET)**

BAC - Starting disk 1 on DU1:

BAC - End of disk 1 on DU1:

BAC - Starting verify pass disk 1 on DU1:

BAC - End of disk 1 on DU1:

BAC - Mount disk 2 on DU1: . Press "RETURN" when done **(RET)**

BAC - Starting disk 2 on DU1:

BAC - End of disk 2 on DU1:

BAC - Starting verify pass disk 2 on DU1:

BAC - End of disk 2 on DU1:

BAC - Mount disk 3 on DU1: . Press "RETURN" when done **(RET)**

BAC - Starting disk 3 on DU1:

BAC - End of disk 3 on DU1:

BAC - Starting verify pass disk 3 on DU1:

BAC - End of disk 3 on DU1:

BAC - Mount disk 4 on DU1: . Press "RETURN" when done **(RET)**

BAC - Starting disk 4 on DU1:

BAC - End of disk 4 on DU1:

BAC - Starting verify pass disk 4 on DU1:

BAC - End of disk 4 on DU 1:

BAC - Completed

This example shows a back up of the fixed disk to several diskettes. You must use the SAVE option with the /IMAGE qualifier when doing a multiple diskette backup operation.

**BACKUP (Cont.)**

```

$ BACKUP/INI/IMAGE:RES/VER DU1: DU0: (RET)

BAC - Mount disk 1 on DU1: . Press "RETURN" when done (RET)

BAC - Starting disk 1 on DU1:

BAC - End of disk 1 on DU1:

BAC - Mount disk 2 on DU1: . Press "RETURN" when done (RET)

BAC - Starting disk 2 on DU1:

BAC - End o f disk 2 on DU 1:

BAC - Mount disk 3 on DU1: . Press "RETURN" when done (RET)

BAC - Starting disk 3 on DU1:

BAC - End of disk 3 on DU1:

BAC - Mount disk 4 on DU1: . Press "RETURN" when done (RET)

BAC - Starting disk 4 on DU1:

BAC - End of disk 4 on DU1:

BAC - Mount disk 5 on DU1: . Press "RETURN" when done (RET)

BAC - Starting disk 5 on DU1:

BAC - End of disk 5 on DU1:

BAC - Completed

```

This example shows a diskette restore operation to the fixed disk. You must specify the RESTORE option on the command line with the /IMAGE qualifier when restoring several diskettes to the fixed disk.

**5.12.4 Messages**

There are instances when you may receive informational, warning, or error messages. Informational messages do not require a response. They give you information such as whether the operation is complete or whether another disk is being verified. Warning messages are usually not fatal, but may represent a severe error. There are also fatal messages which terminate the operation.

This section contains a listing of the BACKUP error messages you may receive, with a description of the problem and the action you should take.

Note that if you cannot find an error message in this section (5.12.4), check Section 5.12.4.1. That section describes and suggests action for error messages (preceded by the mnemonic BRU instead of BAC) you may receive from the BRU utility.

**BACKUP (Cont.)**

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 diskette. This message usually occurs when the output diskette is smaller than the input diskette.

**User Action:** Reinitialize the diskette using the /INDEX:BEGINNING qualifier to BACKUP or INITIALIZE to force the allocation to start at the beginning of the diskette. 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 tape or a disk but not both. Check command for proper device names and reenter the command.

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 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:** Use the ANALYZE/MEDIA command to find and mark any bad blocks on the disk. Then, use BACKUP with the /BAD\_BLOCKS:AUTOMATIC qualifier.



**BACKUP (Cont.)**

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:** Use the ANALYZE/MEDIA command to find and mark any bad blocks on the disk. 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.

#### 5.12.4.1 BRU Messages

This section lists BRU information and error messages, describes the meaning of the message, and suggests corrective action if messages arise when you are using the BACKUP command. A WARNING message indicates an error that may or may not terminate the BACKUP operation. A FATAL message indicates an error that always terminates the operation.

BRU---\*WARNING\*---Allocation failure [directory]filename.typ;ver

**Explanation:** During a copy to a mounted volume, there was not enough free space to copy the specified file.

**User Action:** Create enough free space on the volume and then reenter the command line.

BRU---\*FATAL\*---Allocation for system file exceeds volume limit

**Explanation:** A system file (one of the following files: INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, 000000.DIR) requires more space than is available on the output disk or diskette. This usually occurs if the output diskette is smaller than the input disk.

**User Action:** Use the /POSITION qualifier to force allocation to start at the beginning of the diskette and/or use the /MAXIMUM and /HEADERS qualifiers to reduce the size of INDEXF.SYS.

BRU---\*FATAL\*---Ambiguous option

**Explanation:** An option specified with a qualifier is not unique. For example, the "B" in /POSITION:B could mean either BEGINNING or BLOCK.

**User Action:** Use a form of the option that is unique. All BACKUP options are unique to two characters.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Ambiguous qualifier

**Explanation:** A qualifier is not unique. For example, /RE could mean either /REVISED or /REWIND.

**User Action:** Use a form of the qualifier that is unique. All BACKUP qualifiers are unique to three characters.

BRU---\*WARNING\*---Appending at default bpi on ddnn:  
or

BRU---\*WARNING\*---Appending at 1600 bpi on ddnn:

**Explanation:** The wrong tape density was specified with the /APPEND qualifier. BRU performs an append operation only at the density at which the tape was previously written. The default bpi in the first message is 800.

**User Action:** None. BRU continues at the correct density.

BRU---\*FATAL\*---Attach failed on ddnn:

**Explanation:** BACKUP 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.

BRU---\*FATAL\*---Backup disk read error

**Explanation:** An unrecoverable read error occurred on the output backup disk or diskette, possibly caused by an undetected bad block; or an error occurred while BACKUP was sizing the input or output disk or diskette for a multivolume backup operation.

**User Action:** Use ANALYZE/MEDIA (See Section 5.3) to locate all bad blocks. Then, use BACKUP with the /BAD BLOCKS:AUTOMATIC qualifier to use the existing bad block information on the disk or diskette to create the file BADBLK.SYS.

BRU---\*FATAL\*---Backup disk write error

**Explanation:** An unrecoverable write error occurred on the output backup disk or diskette during a multivolume backup operation. The error could have been caused by an undetected bad block.

**User Action:** See Section 5.3 for information on ANALYZE/MEDIA command. Then, use the /BAD BLOCKS:AUTOMATIC qualifier with BACKUP to use the existing bad block information on the diskette to create the file BADBLK.SYS.

BRU---\*WARNING\*---Bad block data error

**Explanation:** A manually entered bad block location, count, or syntax was incorrect.

**User Action:** Enter the correct information.

## BACKUP (Cont.)

BRU---\*WARNING\*---Bad block file full

**Explanation:** The manual addition of bad blocks has resulted in more than 204(decimal) sets of contiguous bad blocks.

**User Action:** None. You cannot enter more bad blocks than the file will hold. You may not want to use the disk or diskette anymore.

BRU---\*WARNING\*---Block exceed volume size

**Explanation:** You have manually entered a bad block that is larger than the size of the output disk or diskette.

**User Action:** Enter the correct block.

BRU---\*WARNING\*---Boot block is bad

**Explanation:** BACKUP cannot write to the output boot block. Therefore, the output disk or diskette will not be hardware-bootable.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Boot block is corrupt

**Explanation:** The input disk or diskette does not contain a valid boot block. The output disk or diskette will not be hardware-bootable.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Boot block read error

**Explanation:** An error occurred while BACKUP was reading the boot block.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Boot block verify error on ddnn:

**Explanation:** During a backup operation, the boot block on the output device did not match the boot block on the input device.

**User Action:** None. BACKUP continues the operation.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Cannot append on a mounted disk

**Explanation:** A multivolume backup operation to disk or diskette is not possible because you cannot append to a mounted disk or diskette. You can only use the /APPEND qualifier when doing a multivolume disk or diskette backup operation to an unmounted disk or diskette.

**User Action:** You must use another disk or diskette to proceed with your backup operations.

BRU---\*FATAL\*---Cannot mix different types of disks

**Explanation:** The input disk or diskette and the output disk or diskette are of different types for a multivolume disk or diskette operation.

**User Action:** You must specify the same type of disk or diskette for the input and/or output disk or diskette when you do a multivolume restore or backup operation.

BRU---\*WARNING\*---Cannot restore contiguously  
[directory]filename.typ;ver

**Explanation:** The output device does not contain enough contiguous blocks to restore the indicated contiguous file. The file will be restored noncontiguously.

**User Action:** Use the DCL command to make the file contiguous again.

BRU---\*WARNING\*---Close or write attributes error  
[directory]filename.typ;ver I/O ERROR CODE number

**Explanation:** During a copy to a mounted volume, BACKUP encountered an error while attempting to close the specified file.

**User Action:** If possible, determine the cause of the error from the I/O code. If it is correctable, delete the portion of the file that BACKUP has copied, and reenter the command line.

BRU---Completed

**Explanation:** The BACKUP operation is complete.

**User Action:** Enter another BACKUP command line or exit with a CTRL/Z.

BRU---\*FATAL\*---Conflicting qualifiers

**Explanation:** Two or more of the specified qualifiers are mutually exclusive: for example, /SUPERSEDE and /NOSUPERSEDE.

**User Action:** Reenter the command line.

**BACKUP (Cont.)**

BRU---\*WARNING\*---Data ID record verify error

**Explanation:** An error occurred while BACKUP was verifying an 80-byte data-ID record.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Data record verify error [directory]filename.typ;ver  
FILE ID number LBN number

**Explanation:** There was a difference in a data block on input and output devices. The file ID of the file with the error and the LBN of the block follow the message.

If a directory is printed with a file name, the directory is the owner directory from the file's header, not the directory in which the file is contained.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Data was lost due to 10 errors  
[directory]filename.typ;ver

**Explanation:** A tape read error resulted in missing data. The files are restored, but may contain erroneous data.

**User Action:** None. BRU continues operation.

BRU---\*FATAL\*---Device conflict

**Explanation:** Both a tape and a disk drive are specified as part of the input or output device specification.

**User Action:** Reenter the command line, specifying the correct device.

BRU---\*FATAL\*---Device not in system

**Explanation:** A device was specified that does not exist in the system.

**User Action:** Reenter the command line, specifying the correct device specification.

BRU---\*FATAL\*---Device not supported

**Explanation:** The specified device was not a tape, disk, or diskette, or it was a tape, disk or diskette that is not supported by BACKUP.

**User Action:** BACKUP supports only certain devices. Reenter the command line, specifying supported devices.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Directive error

**Explanation:** An internal error occurred in BACKUP.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Directory verify error

**Explanation:** A directory record on the input device did not match a directory record on the output device.

**User Action:** None. BACKUP continues the operation.

BRU---\*---Disk label error

**Explanation:** An I/O error occurred while BACKUP was reading or writing a disk or diskette label. A write error is fatal; a read error is not fatal as long as BACKUP can continue reading the disk or diskette. See the I/O error codes for a definition of the error code number.

**User Action:** If a write error occurred, reenter the command line, specifying a different disk or diskette.

BRU---\*WARNING\*---Disk out of sequence. Please mount correct disk.

**Explanation:** The wrong disk or diskette volume was mounted on the disk or diskette drive during a restore-from-diskette (or disk) operation from a multivolume backup set.

**User Action:** Mount the correct disk or diskette on the drive.

BRU---\*FATAL\*---Disk read error

**Explanation:** An unrecoverable read error occurred on the output disk or diskette, possibly caused by an undetected bad block; or an error occurred while BACKUP was sizing the input or output disk or diskette.

**User Action:** Use ANALYZE/MEDIA (see Section 5.3) to locate all bad blocks. Then, use BACKUP with the /BAD\_BLOCKS:AUTOMATIC qualifier to use the existing bad block information on the disk or diskette to create the file BADBLK.SYS.

BRU---\*FATAL\*---Disk write error

**Explanation:** An unrecoverable write error occurred on the output disk or diskette. The error could have been caused by an undetected bad block.

**User Action:** Use the ANALYZE/MEDIA command to locate all bad blocks (see Section 5.3). Then, use BACKUP with the /BAD\_BLOCKS:AUTOMATIC qualifier to use the existing bad block information on the disk or diskette to create the file BADBLK.SYS.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Doubly defined qualifier

**Explanation:** The same qualifier was specified more than once on the command line.

**User Action:** Reenter the command line, specifying the qualifier once.

BRU---\*WARNING\*---Duplicate blocks found

**Explanation:** A manually entered bad block was already in the bad block file.

**User Action:** None. BACKUP continues the operation.

BRU---End of disk number on ddnn:

**Explanation:** BACKUP has finished transferring data or verifying a disk or diskette.

**User Action:** None. This is an informational message.

BRU---End of tape number on ddnn:

**Explanation:** BRV has finished transferring data or verifying a tape.

**User Action:** None. This is an informational message.

BRU---\*FATAL\*---End of volume encountered. Backup set not found.

**Explanation:** The backup set specified for a restore operation is not on the tape, disk, or diskette volume.

**User Action:** Mount the correct tape, disk, or diskette volume or reenter the command line, specifying the correct backup set name.

BRU---\*WARNING\*---EOT marker error

**Explanation:** During a backup operation, an error occurred while BACKUP was writing or verifying the end-of-tape label on the output tape.

After a restore operation, an error occurred while BACKUP was positioning the tape at the end of a backup set for a subsequent operation.

**User Action:** On a write error, BACKUP rewinds the current tape and places it off line. BACKUP then requests that a new tape be mounted and rewrites the data on a new tape.

On a verify error, BACKUP continues the operation.

On a positioning error, BACKUP finishes the operation. If you want to perform another BACKUP operation on the tape, use the /REWIND qualifier to position the tape at the beginning-of-tape.

**BACKUP (Cont.)**

BRU---\*WARNING\*---Error accessing file I/O ERROR CODE number FILE ID number

**Explanation:** An error occurred while BACKUP was writing data into a file; or BACKUP tried to do a compare read on a file that was already opened. BACKUP will continue with the next file.

See the I/O error codes for a definition of the error code number.

**User Action:** After BACKUP has finished, delete the backup file the error occurred in and then retry the BACKUP operation, specifying only that file.

BRU---\*WARNING\*---Error accessing UFD. Skipping [directory] I/O ERROR CODE number

**Explanation:** During a copy to a mounted volume, an error occurred when BACKUP attempted to access a directory. See the I/O error codes for the definition of the error code number.

**User Action:** If possible, determine the cause of the error from the I/O error code. If correctable, try the copy operation again.

BRU---\*FATAL\*---Error limit exceeded

**Explanation:** During a copy to a mounted volume, an error occurred when BACKUP attempted to access a directory.

**User Action:** If possible, determine the cause of the error from the I/O error code. If correctable, try the copy operation again.

BRU---\*FATAL\*---Error reading command file

**Explanation:** An I/O error occurred while BACKUP was reading the indirect command file.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Error reading data blocks I/O ERROR CODE number FILE ID number LBN number

or  
RECOVERED

**Explanation:** An I/O error occurred while BACKUP was reading a data block from the disk or diskette. The file ID of the file that contains the block and the LBN of the block are displayed as well as the I/O error code. If RECOVERED is printed after the message, the block was recovered by rereading the disk or diskette.

See the I/O error codes for a definition of the error code number.

**User Action:** None. BACKUP continues the operation.



**BACKUP (Cont.)**

BRU---\*WARNING\*---Error reading UFD [directory]

**Explanation:** An I/O error occurred while BACKUP was reading a block from the specified directory. Any files contained in this block of the directory are not backed up.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Error reading UFD header [directory]

**Explanation:** An error occurred while BACKUP was reading the header of the specified directory. Files in this directory are not backed up.

**User Action:** Reenter the command line. If the error still occurs, use VFY to find the lost files.

BRU---\*WARNING\*---Extending index file

**Explanation:** The initial number of file headers was too small. Either 256(decimal) or 16(decimal) more headers will be allocated, depending on the number of blocks on the output disk or diskette.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---Failed to read bad block file

**Explanation:** BACKUP was unable to read the bad block information from a last-track output disk or diskette.

**User Action:** Reenter the command line, using the /BAD\_BLOCKS:OVERRIDE qualifier.

BRU---\*WARNING\*---File header read error [directory]filename.typ;ver  
I/O ERROR CODE number

**User Action:** An I/O error occurred while BACKUP was reading a file header. That file is not backed up. See the I/O error codes for a definition of the error code number.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---File header file verify error  
[directory]filename.typ;ver

**Explanation:** The file header of the specified file on the output device is not the same as that on the input device.

**User Action:** None. BACKUP continues the operation.

**BACKUP (Cont.)**

BRU---\*WARNING\*---File ID area verify error

**Explanation:** The BACKUP-generated file ID area of a data record was different on the input and output devices.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---File ID exceeds maximum number of files

**Explanation:** You specified a maximum number of files with the /MAXIMUM qualifier that was smaller than a file ID encountered on the input volume.

**User Action:** Reenter the command line, specifying a larger value with the /MAXIMUM qualifier.

BRU---\*WARNING\*---File ID sequence number error  
[directory]filename.typ;ver

**Explanation:** Two possible sources of this error are as follows:

1. The sequence number in the file ID of a file does not match the sequence number of the file's entry in the directory.
2. The sequence number of a directory does not match the sequence number of the directory's entry in the Master File Directory (MFD).

In either case, the file or directory is not valid and is not copied.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---File IDs will not be preserved

**Explanation:** File IDs cannot be preserved because the index file bitmap on the output device is too small. This is because the value specified with the /MAXIMUM qualifier was too small.

**User Action:** None. BACKUP continues the operation without preserving file IDs. If your input device had a hardware-bootable system on it, your output device will not be hardware-bootable.

If you want the diskette to be hardware-bootable, perform the BACKUP operation again, specifying a larger value with the /MAXIMUM qualifier.

BRU---\*WARNING\*---File marked for delete [directory]filename.typ;ver

**Explanation:** The marked-for-delete bit (SC.MDL) of the system-controlled characteristics in the file header was set, indicating that the file was partially deleted. The file is not copied.

**User Action:** None. BACKUP continues the operation.

**BACKUP (Cont.)**

BRU---\*WARNING\*---File not found [directory]filename.typ;ver

**Explanation:** During a backup operation, BACKUP cannot find the header for the specified file or directory in the index file. The file is not copied.

During the verify or compare pass of a restore operation, BACKUP cannot find the specified file on the output device.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---File not found

**Explanation:** BACKUP could not find the specified indirect command file.

**User Action:** Reenter the command line, correctly specifying the indirect command file.

BRU---\*WARNING\*---File not superseded [directory]filename.typ;ver

**Explanation:** During a copy to a mounted volume with /NOSUPERSEDE specified (or defaulted), the specified file was not restored because it already existed on the output diskette.

**User Action:** If you want the file to be restored, enter a command line, specifying the file and either /SUPERSEDE or /NEW\_VERISON.

BRU---\*FATAL\*---Handler not resident

**Explanation:** The device driver for the specified device is not loaded.

**User Action:** Load the driver for the specified device or reenter the command line, specifying the correct device name.

BRU---\*WARNING\*---Header ID record is incorrect

**Explanation:** The BACKUP-generated header record on the output device is different from the one on the input device.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Header read error [directory]filename.typ;ver  
I/O ERROR CODE number

**Explanation:** An I/O error occurred while BACKUP was reading a file header in the index file during a backup operation.

If this error occurs during a restore operation, it is fatal.

See the I/O error codes for a definition of the error code number.

**User Action:** None.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Home block read error I/O ERROR CODE number

**Explanation:** An I/O error occurred while BACKUP was reading the home block on the input device. See the I/O error codes for a definition of the error code number.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Home block verify error

**Explanation:** The home block on the output device is different from the home block on the input device.

**User Action:** BACKUP continues, but it is suggested that you retry the operation.

BRU---\*FATAL\*---Home block write error

**Explanation:** An unrecoverable I/O error occurred while BACKUP was writing the home block on the output device.

**User Action:** Use the BAD command to find all the bad blocks on the disk or diskette before initializing it.

BRU---\*FATAL\*---Illegal use of DIRECTORY qualifier

**Explanation:** Three possible sources for this error are as follows:

1. The /DIRECTORY qualifier was specified with an output device.
2. The /DIRECTORY qualifier was specified with a device other than a backup disk or diskette with the BACKUP container file.
3. The /INITIALIZE qualifier or any of its related qualifiers was specified with the /DIRECTORY qualifier.

**User Action:** Refer to the qualifier documentation for the correct use of the qualifiers on your command line.

BRU---\*FATAL\*---Inconsistent INITIALIZE qualifiers

**Explanation:** The /INITIALIZE qualifier or any of its related qualifiers was specified for the output diskette, but the /NOINITIALIZE qualifier was also used.

**User Action:** Reenter the command line.

BRU---\*FATAL\*---Index file header read error I/O ERROR CODE number

**Explanation:** An I/O error occurred while BACKUP was reading the header of the index file on the input diskette. See the I/O error codes for a definition of the error code number.

**User Action:** Reenter the command line.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Index file write error

**Explanation:** An I/O error occurred while BACKUP was writing the index file on the output diskette.

**User Action:** Use the ANALYZE/MEDIA command (see Section 5.3) to identify the bad blocks on the output diskette, then reenter the command line.

BRU---\*FATAL\*---INDEXF.SYS is full

**Explanation:** The index file cannot map any more file headers.

**User Action:** Reenter the command line, specifying a larger value with the /MAXIMUM qualifier.

BRU---\*FATAL\*---Input device equals output device

**Explanation:** The input and output devices must be different.

**User Action:** Reenter the command line, specifying different input and output devices.

BRU---\*FATAL\*---Input line too long

**Explanation:** The maximum length of a command line is 256(decimal) characters.

**User Action:** Truncate qualifiers and options to shorten the line. Make sure the truncated forms are unique. All BACKUP qualifiers are unique to three characters; all options are unique to two characters.

BRU---\*WARNING\*---Input volume structure level differs from output volume

**Explanation:** You have initialized the output volume, specifying with the /MAXIMUM qualifier that the number of files allowed on the volume be greater than 25593(decimal). This causes the index file on the output volume to have more than one file header.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---Internal error

**Explanation:** BACKUP has detected an error within itself. Generally, this should not occur.

**User Action:** Reenter the command line.

BRU---\*FATAL\*---Invalid date or time

**Explanation:** In the command line, a date or time was specified incorrectly or is out of range.

**User Action:** Specify the correct date or time.

**BACKUP (Cont.)**

BRU---\*WARNING\*---Invalid date or time [directory]filename.typ;ver

**Explanation:** An invalid date or time was encountered in a file header during an incremental backup.

**User Action:** None. BACKUP continues the operation. The file is copied.

BRU---\*FATAL\*---Invalid density or tape format

**Explanation:** You specified a density tape that was neither the default bpi (1600) nor 800 bpi; or you attempted to use both 7-track and 9-track tapes in a multivolume tape set.

**User Action:** In the former case, reenter the command line, specifying the correct density. In the latter case, only use all 7-track or all 9-track tapes for a multivolume tape set.

BRU---\*FATAL\*---Invalid disk format

**Explanation:** The disk or diskette that was mounted for an /IMAGE restore operation is not a multivolume backup disk or diskette.

**User Action:** Mount the correct disk or diskette.

BRU---\*FATAL\*---Invalid filename

**Explanation:** The name of the indirect command file is not syntactically correct.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Invalid tape format

**Explanation:** An invalid tape record was read during a restore operation.

**User Action:** None. The invalid record is not restored.

BRU---\*FATAL\*---Invalid value or name

**Explanation:** A value or name specified for a qualifier has illegal syntax or is out of range.

**User Action:** Refer to the descriptions of qualifiers to determine the legal values for the particular qualifier.

BRU---\*FATAL\*---Manufacturer bad sector file is corrupt

**Explanation:** BACKUP was unable to read the bad block information from a last-track output disk or diskette.

**User Action:** Reenter the command line, specifying the /BAD\_BLOCKS:OVERRIDE qualifier.

**BACKUP (Cont.)**

BRU---\*FATAL\*---MFD header read error

**Explanation:** An I/O error occurred while BACKUP was reading the header of the Master File Directory.

**User Action:** Reenter the command line. If the header still cannot be read, the files on the disk or diskette are lost and can be recovered using VFY.

BRU---\*WARNING\*---MFD read error

**Explanation:** An I/O error occurred while BACKUP was reading a block of the MFD. BACKUP cannot copy the directories in that block of the MFD.

**User Action:** Reenter the command line. If the block cannot be read, use VFY to recover the lost files.

BRU---\*FATAL\*---Missing colon

**Explanation:** A qualifier option that accepts a value was not followed by a colon.

**User Action:** Reenter the command line.

BRU---\*FATAL\*---More than 1 level of indirection

**Explanation:** BACKUP does not support more than one level of indirect command files.

**User Action:** Reenter the command line.

BRU---Mount disk n on ddnn:. Press RETURN when done

**Explanation:** This message is issued each time BACKUP requests a disk or diskette for an image backup or restore operation.

**User Action:** Mount the disk or diskette specified on the drive specified and then press RETURN.

BRU---Mount tape n on ddnn:

**Explanation:** There is no tape on the specified drive or the tape is not at load point. This message prints every 2 minutes until the tape is loaded.

**User Action:** Mount the tape specified on the drive specified.

BRU---Mount another disk

**Explanation:** BACKUP is requesting that a new diskette be mounted after encountering a fatal disk or diskette write error.

**User Action:** Mount a new disk or diskette on the drive.

## BACKUP (Cont.)

BRU---Mount another tape

**Explanation:** BACKUP is requesting that a new tape be mounted after encountering a fatal tape write error.

**User Action:** Mount a new tape on the drive.

BRU---\*FATAL\*---Name exceeds maximum allowed length

**Explanation:** A name, such as a backup set name, is longer than 12(decimal) characters. An exception to this rule is during an /IMAGE backup operation to a mounted disk or diskette. In that case, you may specify more than 12(decimal) characters if you are adding the name of the backup set file name to the command line. For additional information, see the documentation describing the qualifiers.

**User Action:** Specify a name not greater than 12 characters.

BRU---\*WARNING\*---No bad block data found

**Explanation:** ANALYZE/MEDIA has not been run on the output diskette to produce a file of the diskette's bad blocks.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---No files found

**Explanation:** During a backup or restore operation, BACKUP did not find any files to transfer.

**User Action:** None.

BRU---\*WARNING\*---Nonfatal qualifier conflicts being ignored

**Explanation:** You entered a qualifier that conflicts with the rest of the command line, but is not fatal if ignored. For example, you used the /REWIND qualifier during a disk to diskette operation.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---No such UFD exists. Skipping [directory]

**Explanation:** During a copy to a mounted volume, BACKUP encountered one or more files in the specified directory on the input volume, but there is no corresponding directory on the output volume.

**User Action:** Reenter the command line, specifying the /DIRECTORY qualifier to create the directory.



**BACKUP (Cont.)**

BRU---\*FATAL\*---Number of headers inconsistent with maximum files

**Explanation:** During an attempt to initialize an output volume, BACKUP found that the maximum number of files specified with the /MAXIMUM qualifier was inconsistent with the number of headers initially allocated to the index file with the /HEADERS qualifier.

**User Action:** See the INITIALIZE command for the legal ranges of values for the /MAXIMUM and /HEADERS qualifier.

BRU---\*WARNING\*---Open error I/O ERROR CODE number FILE I/D number or [directory]filename.typ;ver

**Explanation:** During a copy operation to a mounted volume, an error occurred while BACKUP was attempting to open the specified file. See the I/O error codes for a definition of the error code number.

**User Action:** Determine the cause of the error from the I/O error code. If correctable, delete any portion of the file already copied by BACKUP; then reenter the command line.

BRU---\*FATAL\*---Output disk too fragmented to restore

**Explanation:** The internal tables in BACKUP have overflowed due to the extreme fragmentation of the output disk or diskette. If the output disk or diskette was initialized, then it has an unacceptable number of bad blocks and should not be used as a backup medium.

**User Action:** Use a new disk or diskette as the output device.

BRU---\*FATAL\*---Output device is full

**Explanation:** There are no free blocks on the output disk or diskette. This can occur when the output disk or diskette is smaller than the input disk, or diskette or during an append to a tape or diskette that is already full.

**User Action:** If the output diskette is too small, reenter the command line, specifying only the files you want. If you were doing an append to a tape, disk, or diskette that is already full, reenter the command line, specifying a new tape, disk, or diskette.

BRU---\*FATAL\*---Override invalid with non-last-track device

**Explanation:** The OVERRIDE option may be used only when the output disk or diskette is a last-track device.

**User Action:** Refer to the documentation for correct uses of the qualifiers on your command line.

**BACKUP (Cont.)**

BRU---Please answer yes or no

**Explanation:** BACKUP requires a YES or NO response.

**User Action:** Enter YES or NO at your terminal.\*

BRU---\*WARNING\*---Privilege violation [directory]filename.typ;ver

**Explanation:** During a backup operation, you attempted to copy a file that you do not have read access to.

**User Action:** None. BACKUP does not copy the file.

BRU---\*FATAL\*---Privilege violation

**Explanation:** The mount status of one of the devices is inconsistent with the qualifiers specified in the command line.

**User Action:** Reenter the command line after checking the mount status of the devices and after checking the qualifiers.

BRU---\*FATAL\*---Ran out of spare file IDs

**Explanation:** The output diskette required more file headers than the input diskette, but no free headers were available. The lack of headers is probably due to one of the following reasons:

- The output disk or diskette is too fragmented because of bad blocks.
- There are no free file headers on the input disk or diskette.

**User Action:** If you do not need to preserve file IDs, reenter the command line, specifying the /NOPRESERVE qualifier.

If you want to preserve file IDs, do one of the following:

- If the output disk or diskette is too fragmented, use the ANALYZE/MEDIA command (see Section 5.3) on it to display the number of bad blocks. If this reveals a large number of bad blocks, you may want to use a different disk or diskette.
- Use the DCL command to display the number of free file headers on the input disk or diskette. If there are fewer than four free headers, delete some of the files and then reenter the command line. If you still do not have enough file headers, specify the /NOPRESERVE qualifier in the command line.

BRU---\*WARNING\*---Record not expected size

**Explanation:** The record read on the output device during a verify or compare operation was not the expected size.

**User Action:** None. BACKUP continues the operation.

**BACKUP (Cont.)**

BRU---\*FATAL\*---Required input device missing

**Explanation:** The input device was not specified on the command line or in response to the prompt.

**User Action:** Reenter the command line.

BRU---\*FATAL\*---Required output device missing

**Explanation:** The output device was not specified on the command line or in response to the prompt.

**User Action:** Reenter the command line.

BRU---Rewind error on ddnn:

**Explanation:** An I/O error occurred during a tape rewind. This error is fatal if it occurs on the first tape of a tape set or during a rewind for a verify operation. The error is not fatal if BACKUP is rewinding a tape it is finished with.

**User Action:** If the error is fatal, reenter the command line. If the error is not fatal, no action is required.

BRU---\*FATAL\*---Search for home block failed

**Explanation:** The home block could not be found on the input disk or diskette. Either the home block is bad, or the disk or diskette is not in Files-11 format.

**User Action:** Check to see that you have the correct disk or diskette.

BRU---Starting tape n on ddnn:

**Explanation:** This message tells you which tape is being copied to or from which drive.

**User Action:** None. This is an informational message.

BRU---Starting verify pass

**Explanation:** This message tells you that the verify pass of a diskette-to-diskette (or disk) operation is beginning.

**User Action:** None. This is an informational message.

BRU---Starting verify pass tape n on ddnn:

**Explanation:** This message tells you which tape is being verified during a backup or restore operation.

**User Action:** None. This is an informational message.

## BACKUP (Cont.)

BRU---\*FATAL\*---Syntax error

**Explanation:** The command line is invalid.

**User Action:** Reenter the command line.

BRU---Tape label error on ddnn: I/O Error code number

**Explanation:** An I/O error occurred while BACKUP was reading or writing a tape label. A write error is fatal; a read error is not fatal as long as BACKUP can continue reading the tape.

**User Action:** If a write error occurred, reenter the command line specifying a different tape.

BRU---\*WARNING\*---Tape label verify error

**Explanation:** BACKUP detected an error in the tape label of the input or output tape volume during a verify operation.

**User Action:** None. BACKUP continues the operation.

BRU---\*WARNING\*---Tape not at BOT. No rewind or append specified

**Explanation:** For a backup operation to tape, BACKUP will not process a tape that is not at BOT unless the /APPEND qualifier was specified.

**User Action:** If you want to start writing at the beginning of the tape, use the /REWIND qualifier.

You can append to tape only at the end of the last backup set. If the tape is already positioned there, specify the /APPEND qualifier in the command line. If it is not positioned at the end of the last backup set, specify both the /REWIND and /APPEND qualifiers in the command line.

BRU---\*WARNING\*---Tape out of sequence. Please mount the correct tape.

**Explanation:** The wrong tape volume was mounted on the tape drive during a restore operation from a tape backup set.

**User Action:** Mount the correct tape on the drive.

BRU---\*WARNING\*---Tape positioning error. Backspace failed.

**Explanation:** During a backup operation, the tape was not positioned properly for a future append operation.

**User Action:** Rewind the tape before attempting the append operation.

## BACKUP (Cont.)

BRU---\*FATAL\*---Tape positioning error. No EOF encountered. I/O error code number.

**Explanation:** The tape spacing operation failed to find the end of the tape volume for an append operation.

**User Action:** Reenter the command line.

BRU---\*WARNING\*---Tape read error

**Explanation:** An I/O error occurred while BACKUP was reading a tape.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---Tape to tape not supported

**Explanation:** BACKUP does not back up a tape to another tape.

**User Action:** None.

BRU---\*WARNING\*---Tape write error I/O error code number

**Explanation:** An I/O error occurred while BACKUP was writing to tape. BACKUP rewinds the tape and then requests that another tape be mounted.

**User Action:** If the error is related to the tape drive, terminate the BACKUP operation and begin again from another tape drive.

BRU---This disk will not contain a hardware bootable system

**Explanation:** The output disk or diskette will not be hardware-bootable. This can be caused by the following:

- The input disk or diskette not being bootable
- The system image not being copied
- Copying to a disk or diskette of different size or type

### NOTE

This message is not issued when BACKUP is restoring to a mounted volume.

**User Action:** None.

## BACKUP (Cont.)

BRU---\*FATAL\*---Too many devices

**Explanation:** For a conventional backup a disk, or diskette may be specified only once as an input or output device. However, up to eight tape, disk, or diskette drives may be used during an /IMAGE operation.

**User Action:** Reenter the command line, specifying the correct number of drives.

BRU---\*FATAL\*---Too many file specifications

**Explanation:** More than 16(decimal) file specifications were specified on the command line.

**User Action:** Reenter the command line. You can use wildcards to reduce the number of file specifications on the command line.

BRU---\*FATAL\*---UFD or MFD requires unsupported extension headers

**Explanation:** BACKUP does not support extension headers for MFDs or directories.

**User Action:** This error should not occur. Please reenter the command line.

BRU---\*WARNING\*---UFD record verify error

**Explanation:** There is a difference between input and output devices on a directory record.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---Unknown option

**Explanation:** An option was specified that was not recognized by BACKUP.

**User Action:** Check your qualifiers and options and reenter the command line.

BRU---\*FATAL\*---Unknown qualifier

**Explanation:** A qualifier was specified that was not recognized by BACKUP.

**User Action:** Check the qualifiers and reenter the command line.

BRU---\*FATAL\*---Unsupported structure level

**Explanation:** The file structure level on the input diskette is not supported by BRU.

**User Action:** Check that you have the correct diskette. (See the descriptions of the /HEADERS and /MAXIMUM qualifiers.)

**BACKUP (Cont.)**

BRU---\*WARNING\*---LBN not in file

**Explanation:** A file ID was encountered that is larger than the maximum file ID in the index file. The file is ignored. This error message occurs if a directory entry was corrupted on the input diskette.

**User Action:** None. BACKUP continues the operation.

BRU---\*FATAL\*---Verify lost

**Explanation:** During the verify pass of a disk-to-tape backup operation, BACKUP lost synchronization between the input and the output. This is usually caused by the tape position being lost or by backing up from a disk that is mounted and then changed during the backup operation.

**User Action:** Reenter the command line.

BRU---\*FATAL\*---Volume not a backup disk

**Explanation:** The disk or diskette mounted for an append or restore operation does not contain a backup set file generated by BACKUP.

**User Action:** Check to see that you have the correct disk or diskette and reenter the command line.

BRU---\*FATAL\*---Volume not a backup tape

**Explanation:** The tape mounted for an append or restore operation was not generated by BACKUP, or the tape is not positioned correctly for an append operation.

**User Action:** Check to see that you have the correct tape, or reenter the command line specifying the /REWIND qualifier to position the tape.

BRU---\*FATAL\*---Device not ready

**Explanation:** The device is not on line.

**User Action:** Put the device on line and reenter the command line.

BRU---\*FATAL\*---Volume write-locked

**Explanation:** The output device is not write-enabled.

**User Action:** If the output device is a tape, insert a write ring to make it write-enabled. If it is a disk or diskette, press the write enable switch on the disk or diskette drive.

## BACKUP (Cont.)

BRU---\*FATAL\*---Wrong backup set

**Explanation:** During a restore operation from a multivolume backup set, BACKUP found that one of the tapes, disks, or diskettes does not contain the correct backup set.

**User Action:** Reenter the command line, specifying the correct tapes, disks, or diskettes.

BRU---\*FATAL\*---Wrong input volume label

**Explanation:** The input volume label specified with the /INVOLUME qualifier does not match the volume label of the input device.

**User Action:** Reenter the command line, specifying the correct input volume label.

### 5.13 SET AND SHOW

You can display assignments, and you can also set and display certain device characteristics.



## SHOW ASSIGNMENTS

## SHOW LOGICAL

### 5.13.1 SHOW ASSIGNMENTS and SHOW LOGICALS

SHOW ASSIGNMENTS (SHOW LOGICALS displays the same information) displays at your terminal all local and login logical assignments. Privileged users can display assignments for other terminals as well as all assignments in the operating system.

Logical assignments are established by the ASSIGN, DEFINE, and SET DEFAULT commands, and by ACNT, the Account File Maintenance Program. See the RSX-11M/M-PLUS System Management Guide for more information on ACNT.

#### Format

```
SHOW ASSIGNMENTS[/qualifier]
```

or

```
SHOW LOGICALS[/qualifier]
```

#### Command Qualifiers

```
/ALL
/GLOBAL
/GROUP[:g]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:
```

#### Parameters

None.

#### Command Qualifiers

```
/ALL
```

Displays all of your local, login, and group logical name assignments, as well as all global assignments.

You can also use this qualifier with the /TERMINAL:ttnn: qualifier to see the local, login, and group logical names for terminal ttnn: displayed on your terminal.

```
/GLOBAL
/SYSTEM
```

Specifies that all global logical assignments in the operating system are to be displayed on your terminal. The qualifier /SYSTEM is a synonym for /GLOBAL. These qualifiers are privileged.

## SHOW ASSIGNMENTS

### SHOW LOGICALS (Cont.)

/GROUP[:g]

Displays the group logical assignments for users with the specified User Identification Code (UIC) group number, g.

Nonprivileged users can see the group logical assignments of their own group. In this case, you can either specify /GROUP without an argument or use your own group number as the argument.

Only privileged users can see the logical assignments of other groups.

This qualifier is valid only if your system supports extended logical names.

/LOCAL

Specifies that local and login logical assignments for your terminal are to be displayed on your terminal. This is the default.

/LOGIN

Specifies that local and login logical assignments for your terminal are to be displayed on your terminal. This is a privileged qualifier.

/TERMINAL:ttnn:

Specifies that local and login logical assignments for terminal ttnn: are to be displayed on your terminal. This is a privileged qualifier.

You can also use this qualifier with /ALL to see all of the logical assignments for terminal ttnn:.

#### Examples

```
$ SHOW (RET)
Function? ASSIGNMENTS (RET)
LP0:      DU1:      LOCAL  TI - TT1:
TP0:      TT1:      LOCAL  TI - TT1:
SY0:      DY1:      LOGIN   TI - TT1:
```

This example (only for systems without extended logical name support) shows the display of your local and login logical assignments. The logical name (terminated by a colon) is displayed first, followed by the device name. Finally, the type of assignment and your terminal number are given.

You can use these logical names in place of device specifications. Output to LP0: will actually go to DU1:. Output to TP0: will go to TT1:, which is TI:. In addition, the user has the login assignment of SY0: to DY1:. This means that DY1: is the default device for this user.

## SHOW ASSIGNMENTS

### SHOW LOGICALS (Cont.)

```
$ SHOW (RET)
Function? ASSIGNMENTS (RET)
LP0 = DU1: (Local, TT:)
INFILE = DU0:[DANIEL]ADDRESS.TXT (Local, TT:)
@@PE&& WISS = STRANGE string (Login, TT:)
SYS$LOGIN = DU0: [SAMUAL] (Login, Final, TT:)
```

This example shows the display of your local and login logical assignments. The logical name is displayed first, followed by the equivalence name. Finally, the type of assignment and your terminal number are given.

You can use these logical names in place of device and file specifications. References to LP0 will actually go to DU1:. The file ADDRESS.TXT, which is located on device DU0: in directory [DANIEL], is an input file for a task that specifies the logical name INFILE. In addition, the string "STRANGE string" is the translation of the logical name "@@PE&& WISS". Finally, the user has given the login assignment of SYS\$LOGIN to DU0:[SAMUAL]. This means that DU0:[SAMUAL] is the default device and directory for this user.

```
$ SHOW ASSIGNMENTS/LOCAL (RET)
```

This example is the equivalent of the previous example.

```
$ SHOW ASSIGNMENTS/ALL (RET)
WK = LB: (Global, Final)
TEXT = DU0:[SYSTST]LOGIN.TXT (Local, TT:)
SYS$LOGIN = DU: (QUERY) (Login, Final, TT:)
```

This example displays all the logical assignments (local, login, global, and system) for your terminal. The system has given the global logical name WK to the pseudo device, LB: and the login logical name SYS\$LOGIN to the directory, DU0:[QUERY]. The user has given the local logical name TEXT to the file specification, DU0:[SYSTST] LOGIN.TXT.

```
$ SHOW ASSIGNMENTS/TERMINAL:TT3: (RET)
MP0 = SY0: (Local, TT3: )
```

This example, which requires a privileged terminal and a terminal (TT3:) that is logged in, displays the logical assignments for terminal TT3:. This user has given the local logical name MP0 to the pseudo device, SY0:.

```
$ SHOW ASSIGNMENTS/GLOBAL (RET)
WK = LB: (Global, Final)
IN0 = SY0: (Global, Final)
EX0 = SY0: (Global, Final)
```

This example, which requires a privileged terminal, displays all global logical names.

## SET DEVICE

### 5.13.2 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; no new tasks will be checkpointed to the file.

## 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.

Chapter 4, 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 15 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 either the SET DEVICE/WIDTH:n command or the SET TERMINAL/WIDTH:n command.

### Examples

```
$ SET (RET)
Function? DEVICE (RET)
Device? LP10 (RET)
Attribute? WIDTH:80 (RET)
```

This example sets the I/O buffer size (line length) to 80 (decimal) characters on the line printer LP0:.

## SET DEVICE (Cont.)

\$ SET DEVICE:LP0:/LOWERCASE (RET)

This example sets LP0: 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:DL3:/CHECKPOINT\_FILE:256 (RET)

This example creates a checkpoint file of 256 (decimal) blocks on DL3:.

## Note

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 the command using the 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 or RL01s.

**SHOW DEVICES****5.13.3 SHOW DEVICES**

SHOW DEVICES displays information about the devices included in the system.

**Format**

```
SHOW DEVICES[/attribute]
```

**Device Attributes**

```
/dd[nn]:  
/[NO]PUBLIC  
/WIDTH:ddnn:  
/[NO]SYSTEM (synonym for /PUBLIC)
```

**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.

```
/dd[nn]:
```

Displays information about all devices of a particular type on the system. 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. Because the same controller may control devices whose disks are not interchangeable, a generic allocate (ALLOCATE DU) may allocate a device which cannot accept your disk. See examples.

```
/[NO]PUBLIC  
/[NO]SYSTEM
```

Displays a list of all devices set (or not set) PUBLIC. The /[NO]SYSTEM qualifier is a synonym included for VAX/VMS compatibility.

```
/WIDTH:ddnn:
```

Displays the size of the I/O buffer (line length) for a particular device, including a terminal.

**Display Information**

The display from SHOW DEVICES can include a number of messages.

**SHOW DEVICES (Cont.)**

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).

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, RL02, RX02. If you want 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 cannot 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 to the terminal.

LOADED

Indicates that a loadable device driver is currently loaded.



## SHOW DEVICES (Cont.)

## UNLOADED

Indicates that a loadable device driver is currently not loaded.

## SPOOLED

Indicates that a device is under the control of the Queue Manager.

## 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
DR0:      TT3: - PRIVATE MOUNTED LOADED TYPE=RM03
DR1:      PUBLIC MOUNTED LOADED TYPE=RM05
DU0:      PUBLIC MOUNTED LOADED TYPE=RM80
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
NLO:     LOADED
VTO:     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 current directory is shown. 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 (Cont.)**

```
$ SHOW DEVICES DU: (RET)
DU0:    PUBLIC MOUNTED LOADED LABEL=MICRORSX TYPE=RD51
DU1:    PUBLIC MOUNTED LOADED TYPE=RD51
DU2:    PUBLIC LOADED TYPE=RX50
DU3:    OFFLINE LOADED TYPE=unknown
```

This example displays information about all devices of the type DU:. The mnemonic (DU<sub>nn</sub>;) identifies the device controller and the model name (RD51) indicates the physical device type.

```
$ SHOW DEVICES/PUBLIC (RET)
PUB=DU0:
PUB=DU1:
PUB=DU2:
PUB=LP0:
```

This example displays information about all public devices. Public devices are accessible to all users. They may be mounted by any users to assure continued access to the volume mounted on the device.

```
$ SHOW DEVICE: LP0:/WIDTH (RET)
BUF=LP0:00132.
$ SHOW DEVICE:TT11:/WIDTH (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

### LINK AND LIBRARY COMMANDS

Program development on RSX-11M-PLUS 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-PLUS.

To create a source program, you must be familiar with editing. You must also know of one of the programming languages available on your system. RSX-11M-PLUS programming languages are available separately as software options. This chapter describes only the LINK and LIBRARY commands and how they apply to the program development process. Other commands for program development are described in the documentation supplied with your language.

RSX-11M-PLUS 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.

#### 6.1 LINKING YOUR TASK WITH THE TASK BUILDER

The following describes both the Task Builder and the LINK command that invokes the Task Builder. The Task Builder is a large, complex RSX-11M-PLUS utility program that converts a file that has been assembled or compiled into an executable task. Tasks can be coded in many complicated ways and they can be built by the Task Builder in additionally complex ways. This chapter describes, for novices only, the most simple functions of the Task Builder and the LINK command used to invoke it.

If you are an experienced RSX programmer, you may find this chapter too fundamental and simple. In this case, refer to the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.

If this RSX-11M-PLUS system is your first exposure to an RSX system and you are not clear about how to create a task on it, read on.

## LINK AND LIBRARY COMMANDS

After reading the following sections on the Task Builder and the LINK command, you should understand the process of the development of a task from code to a completely built task. This description assumes that you know what a program is and how to assemble or compile it. It also assumes that you know about some fundamentals of computers, such as memory, disks, disk directories, and terminals.

### 6.2 REVIEW OF THE PROGRAMMING PROCESS

This section reviews the steps you must take from writing a program to creating a task. It also discusses the files that are created in this process.

#### 6.2.1 The Steps You Take When Programming

Programming is a process that has several major steps or parts. Here the discussion includes only those steps that you take to create a running task. Other steps in the programming process, such as debugging and testing, are not discussed.

The three major steps in programming are as follows:

1. Writing the program
2. Assembling or compiling the program
3. Creating a runnable task from the program

The first step in the programming process is writing the program (code) in one of a number of source languages. A source language is a specific collection of defined symbols, coded phrases, and characters. The source language may resemble closely the native, binary machine language. If it does, it is called an assembly language. If the source language more closely resembles English (or your native tongue), then it is called a high-level language. An example of an assembly language is DIGITAL's MACRO-11 assembly language. Two examples of high-level languages are FORTRAN-77 and COBOL-81. You enter the source language into a file by typing the code on a terminal. The file that you create in this way is the input file for the next step.

The second step in the programming process is assembling or compiling the program created in the first step. If you wrote the code in assembly language, you use DIGITAL's MACRO-11 assembler to assemble the code you have written. The assembler converts the program you have written into a binary language (object code) that the system understands. If you wrote the code in a high-level language, you use a compiler to create a binary language (object code) that the system understands. With either the assembler or compiler, you use the system to create object code, which is a machine-usable and readable binary language. However, the object code cannot yet execute on the machine as a task that can perform useful work. To make the object code usable as a task, you must use the Task Builder, which is briefly described in the next step.

The third step in the programming process is creating a task that does useful work from the object code created in the second step. A task is also a binary language readable by the system. However, a task differs from object code in that the task knows all the addresses of areas to which it will refer, it has its own addresses arranged in order, and it has special information attached to it for use by the system. You create a task by using the LINK command, which invokes

## LINK AND LIBRARY COMMANDS

the Task Builder. The Task Builder uses the object code in the object files that you created in the second step and transforms them into a runnable task.

Discussed in the following sections is a brief description of the files that you will use and create when you program with RSX-11M-PLUS.

### 6.2.2 The Source File

The source file is a file that you create when you write your program using a specific programming language. For example, you create a file with a file type of .MAC when you use the MACRO-11 Assembly Language. For another example, you create a file with the file type of .FTN when you use the FORTRAN-77 language. Whatever language you use, the source file must be assembled by the assembler or compiled by the compiler before you continue. The assembler or compiler produces files called object files.

### 6.2.3 The Object File

The object file is a file that has the .OBJ file type. It contains binary code and is not directly readable. If you try to print it, the file may cause excessive form feeds on the printer and will appear as a confused collection of meaningless characters and spaces. If you try to type the object file on the screen, the file will appear as a meaningless arrangement of spaces and random characters and may lock the keyboard or screen. The object file is the file produced by the assembler or compiler that operated on your source code and does not, therefore, have the proper code for printing on the printer or displaying on the terminal. The object file is designed to be changed into a task. By itself, the object file cannot execute as a task. To create a task, the Task Builder must be used.

### 6.2.4 The Task File

The task file is a file that has a .TSK file type. It contains binary code that can run as a task and do useful work on the RSX-11M-PLUS system. The Task Builder takes object code produced by the assembler or compiler and adds structure to it so that the task can execute properly. The task file can be printed and displayed, but only with the same confusing results as when trying to print or display the object file.

## 6.3 INPUT FILES FOR THE TASK BUILDER

The Task builder accepts and uses four types of input files:

1. Object files
2. Library files
3. Overlay description files
4. Command files

These files are described next.

### 6.3.1 Object Files as Input to the Task Builder

The assembler or compiler processes the source code that you have written and produces the object file. The assembler produces an object file from assembly language source code. The compiler produces an object file from high-level language source code. An object file contains nondisplayable binary code. The object file must be named and must be a discrete part of the necessary code needed to create an executable task. The object file is then called a module (an object module). A task may contain one or more modules, each of which contains a separate logical part of the task. Separate modules are used for the following reasons:

- Separate modules may be written by different persons.
- Separate modules may contain discrete logical parts of a task. For example, a task that does calculations may have an add module, a subtract module, a multiple module, and so forth.
- The same module may be used by different tasks though included at separate times for those tasks; therefore, it need be coded only once. For example, many tasks can do addition and use the same add module if the module and tasks are designed properly.
- A task divided into logically discrete parts makes the task easier to understand and debug.
- A module may contain coded routines, each of which is usable by several tasks. A module coded in this way is called a library. Section 6.3.2 discusses libraries. See also the discussion of the LIBRARY command in Section 6.6.

The Task Builder accepts one or more object module files from a disk and creates a task from them.

When you use the LINK command, described in a later section, you can specify one or more object files as input to the Task Builder. Object files have an .OBJ file type. The following example will begin to familiarize you with the required form of the LINK command when you use object files as input to the Task Builder:

```
$ LINK/TASK MODULE1.OBJ,MODULE2.OBJ,MODULE3.OBJ  (RET)
```

This LINK command shows three object modules as input to the Task Builder. The Task Builder combines them to create one task named MODULE1.TSK. You would run this task by issuing the command RUN MODULE1. You need not specify .OBJ as a file type in the LINK command because the Task Builder assumes that an input file has an .OBJ file type. However, the file must actually have this file type.

### 6.3.2 Library Files as Input to the Task Builder

A library file is another kind of object file and it has an OLB file type. However, a library file is coded with a special purpose in mind. What the library file contains is coded routines that may be commonly used by your task and other tasks. Your task would then call the routines in the library file when they are needed. This saves a lot of time and effort because, in many applications for specific business sites, many routines are useful for a wide range of programs. These routines may include, for example, specific kinds of calculating routines such as addition or multiplication, or certain kinds of graphic charts. The DCL LIBRARY command is used for maintaining libraries (see Section 6.6).

## LINK AND LIBRARY COMMANDS

The LINK command has two ways of specifying a library file in the command line. The first is shown next as an example. The two ways are more fully described in a later section. Here is a LINK command that uses an entire library module:

```
$ LINK/TASK TASKMOD.OBJ,LIBMOD.OLB/LIBRARY (RET)
```

This LINK command creates a task from two input files. The first input file is TASKMOD, which is an ordinary object file. The second input file is LIBMOD, which is a library file. In this example, the TASKMOD file refers to and uses the LIBMOD library. The LIBMOD.OLB file has a special /LIBRARY qualifier on it. The qualifier denotes that LIBMOD is a library. With this command, the routines in the library that the task references become part of the task.

The .OLB file type used in the previous LINK command is understood to be present because of the /LIBRARY qualifier; therefore, the default file type for a library used in this way is .OLB. That is, the .OLB file type need not be specified in the LINK command line, but the file must actually have those file types.

### 6.3.3 Overlay Description Files as Input to the Task Builder

The overlay description file is a special file unlike the object files that have been previously discussed. It contains special mathematical-like language that contains the names of object files and libraries in an algebraic relationship. This file has the .ODL file type.

It is best to mention here that this chapter does not describe overlays at all; nor does it describe overlaid tasks in great detail. Overlays are an advanced topic for programmers, and complete information is available in the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual. However, a simplified description of an overlaid task follows:

An overlaid task is one in which parts of the task are not always together in memory. Some parts of the task may be on the disk and some in memory at different times during task execution. When a part of the task that is on disk is called into memory, it may be read into memory over another part of the task already in memory. This action and interaction between disk and memory and parts of tasks can get quite complex. That is the reason for the overlay description file.

The overlay description file (.ODL file) describes the relationship between the parts of the task and libraries in such a way that everything works correctly at the right time during task execution. The only file that needs to be specified when you build an overlaid task is the overlay description file. The overlay description file contains all the necessary logical relationships and module names for

the task to build correctly. Therefore, you may see a LINK command that looks like one of the following:

```
$ LINK/TASK INPUTFILE.ODL/OVERLAY_DESCRIPTION (RET)
```

or

```
$ LINK/TASK/OVERLAY_DESCRIPTION INPUTFILE (RET)
```

These two LINK commands specify that a task is to be built using a single input file that describes an arrangement of overlays for the task. If you use the /OVERLAY\_DESCRIPTION qualifier, the Task Builder assumes the .ODL file type. No other files need to be mentioned as input files because all the other files are designated in the .ODL file. The /OVERLAY DESCRIPTION qualifier can be in either part of the LINK command; that is, to the left of the blank space as a command qualifier, or appended to the input file as an input file qualifier. The /OVERLAY DESCRIPTION qualifier specifies that the single input file describes how the task is to be built and overlaid. The .ODL file type is understood to be the default file type if you specify /OVERLAY\_DESCRIPTION. Therefore, you need not specify .ODL in the command line in this case, even though the file must actually have that file type.

#### 6.3.4 Command Files as Input to the Task Builder

The command file is another special file unlike all the other Task Builder input files. The command file contains Task Builder commands in a special language and syntax only usable by the Task Builder. This file contains all the commands needed to build a task. Because the Task Builder is large and complex, the command file may contain many commands, qualifiers, and options not mentioned in this simple Task Builder and LINK command description. However, you need not know how to create a Task Builder command file at this time. The only knowledge you require at this point is how to use the LINK command to build a task by using a command file. The following example illustrates this:

```
$ LINK @BUILDFIL.CMD (RET)
```

Command files have a CMD file type. The file type need not be used in the LINK command because CMD is understood to be the default file type. However, the file must actually have a CMD file type. The at sign (@) designates that the system is to process the command file and send the commands that it contains to the Task Builder.

#### 6.4 THE TASK BUILDER FUNCTIONS

After you have coded a program in a programming language and then compiled it, the result is an object file; this file has an .OBJ file type. However, a large program can be coded and compiled in one or more discrete sections or parts called modules. When this is done, all the modules should have a unique name. Every programming language has a statement in it that you can use to name the module. Also, all the modules will probably have defined unique symbols. For example, these symbols may be labels of routines in the code. The work that



remains is the job of the Task Builder. Simply explained, the Task Builder does the following:

1. Puts the modules in a single unit called a task (with a TSK file type) that has a logical order of code.
2. Arranges the addresses of the combined modules in sequential order.
3. Assigns addresses to symbols and makes any global symbols known to the entire task. Not all symbols are global ones; some may need to be known only to a single module.

## 6.4.1 Creating a Task

The Task Builder uses files that have the .OBJ file type to create a task. The resulting task has a TSK file type. The .OBJ files are files that have binary coded information arranged in certain ways. These files are not printable or readable on a terminal. If you were able to see one, it would be a very long string of binary numbers like these:

```
0110001101011100011100010101110100010100000111
```

An object file cannot be executed by a computer system. The Task Builder takes one or more object files and arranges them into a single unit of binary code called a task. The Task Builder also affixes to the task certain structures that are used by the RSX-11M-PLUS system code.

The LINK command, discussed in the next section, is the command you use to invoke the Task Builder and create a task from one or more object modules. However, for now, the following is an example LINK command line that creates one task from three modules:

```
$ LINK/TASK:MYCODE MODULE1,MODULE2,MODULE3 (RET)
```

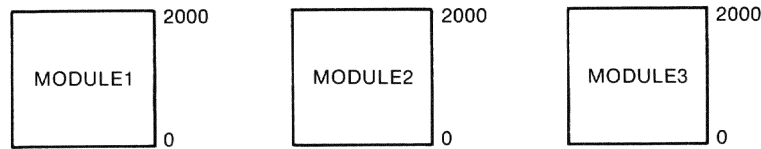
Here, the LINK command line causes the Task Builder to create a single task called MYCODE from three modules. The modules are separately coded and were named differently. However, the entire program could have been coded as a single module and the single object module would have only one name.

## 6.4.2 Arranging Addresses

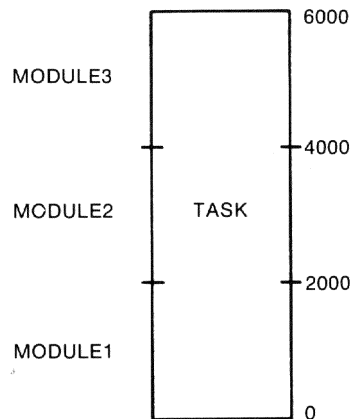
Every unique module that you code has addresses ranging from 0 through the highest address in the module. In other words, the beginning of the code in each module starts at 0 and ends at the highest address. If a task is to be constructed out of several modules, each module will have different information at the same numeric address. The program cannot work this way. Therefore, when combining the modules, the Task Builder assigns new addresses to all the module statements. This assignment results in a single task that has a single address of 0 at the beginning. The addresses are then sequential from that point through the end of the task. Figure 6-1 shows the relationships among addresses for a 3-module task.

## LINK AND LIBRARY COMMANDS

In Figure 6-1, three modules of equal size (2000 octal words) have been compiled. Each module has addresses starting from 0 and continuing through 2000. After the three modules have been made into a task by the Task Builder, the modules are combined and the addresses reassigned. Figure 6-1 represents an ideal picture of the combination of modules into a task. The actual size of a task created from these modules would be larger because of structures added by the Task Builder.



3 SEPARATE MODULES OF 2K-WORDS EACH



TASK OF 3 MODULES CREATED BY THE TASK BUILDER

ZK-1371-83

Figure 6-1  
Module Address Assignment

### 6.4.3 Finding Symbols and Their Definitions

After arranging separate modules into a continuous task, the Task builder must find all symbols and all locations that reference the symbols. It then assigns the correct address in the locations that reference the symbols.

For example, in a task that contains three modules, module 1 may reference a symbol in module 3. Module 1 cannot contain a correct address for the symbol in module 3 until the modules are combined by the Task Builder and the Task Builder looks for the symbol in module 3 and determines its location. To do this work, the Task Builder constructs many internal tables while building the task and a special output file.

## LINK AND LIBRARY COMMANDS

One of the results of symbol resolution by the Task Builder is the creation of an output file called the symbol definition file. It has an .STB file type. It is important that you specify that this file be constructed by the Task Builder when you use the LINK command to create a task. This file is sometimes needed for certain kinds of tasks. The need for this file and possible exceptions are beyond the scope of this chapter. A full description is available in the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.

## LINK

### 6.5 THE LINK COMMAND

The LINK command invokes the Task Builder, a system utility that builds the tasks, 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. Thus, 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. The following command invokes the Task Builder:

```
$ LINK (RET)
File(s)?SAMPLE (RET)
```

This command produces a single output file called SAMPLE.TSK. The return of the dollar sign prompt (\$) tells you that the task build completed successfully.

If, in addition to the task image file, you 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.

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.

## LINK (Cont.)

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.

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)
```

## LINK (Cont.)

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 two 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 preceeding 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/OPTIONSBIGJON (RET)
Option? @SPARKY.CMD (RET)
Option? (RET)
$
```

or

```
$ LINK/MAP/OPTIONS:@SPARKY.CMDBIGJON (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.
3. DCL places the TKB-format command in a TKB indirect command file and passes the indirect command file to the Task Builder.

## LINK (Cont.)

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 by using the defaults and by following the examples given in the command description (Section 6.5.2), 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. It explains all the options, defines all the terms used in the TKB environment, and 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 LINK command works 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 want to repeat a particular task build. See the discussion of the /SAVE qualifier in Section 6.5.1.

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 one 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 output file type STB is called a symbol definition file and it takes the third position in the list of TKB output files to the left of the equal sign.

In TKB format, you can alter the output file by adding switches (consisting of a slash and two letters) to each output file specification.

For example, 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 this 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.

## 6.5.1 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])
        CLI
        DATA_SPACE
        EAE
        FAST_MAP
        FPP
        PIC
        POSITION_INDEPENDENT
/COMPATIBLE
/CROSS_REFERENCE
/DEBUG[:filespec]
/[NO]EXECUTABLE[:filespec]
/ERROR_LIMIT:n
/[NO]EXTERNAL
/FAST
/FULL_SEARCH
/[NO]HEADER
/[NO]IO_PAGE
/LONG
/MAP[:filespec]
/[NO]MEMORY_MANAGEMENT[:n]
```

**LINK (Cont.)**

```

/OPTIONS[:@optionsspec]
/OVERLAY_DESCRIPTION
/POSTMORTEM
/[NO]PRINT
/PRIVILEGED[:n]
/[NO]RECEIVE
/[NO]RESIDENT_OVERLAYS
/SAVE
/[NO]SEGREGATE
/SEQUENTIAL
/SHAREABLE[:arg]
        COMMON
        LIBRARY
        TASK
/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.

## LINK (Cont.)

In TKB format, apply the /AC:n switch to the .TSK file specification.

```
/[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 an important reason for not doing so.

You can specify how the checkpointing of the task is handled through the arguments to the /CHECKPOINT qualifier. For example, 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 only run one copy of such tasks.)

The following list demonstrates how the checkpointability of a task can have an impact on the efficient operation of the entire system, and 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.

**LINK (Cont.)**

In TKB format, the following qualifiers require specific switches for each task image file:

/CHECKPOINT:SYSTEM	apply the /CP switch to the task image file
/CHECKPOINT:TASK	apply the /AL switch to the task image file
/NOCHECKPOINT	apply the /-CP switch to the task image file

/CODE:(arg[,s])

CLI  
DATA\_SPACE  
EAE  
FAST\_MAP  
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.

The CLI argument informs the Task Builder that the task is a CLI task. In TKB format, apply the /CLI switch to the .TSK file specification.

The DATA\_SPACE argument specifies that the task employ user-mode instruction- and data-space. In TKB format, apply the /ID switch to the .TSK file specification.

The EAE argument specifies that the task use 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 FAST\_MAP argument enables the task to use the Fast Mapping feature of the Executive. In TKB format, apply the /FM switch to the .TSK file specification.

The FPP argument specifies that the task use the Floating Point Processor. This hardware is optional and may not be part of your system. In TKB format, apply the /FP switch to the .TSK file specification.

The PIC argument specifies that the resident common or library being built is to be position-independent. In TKB format, apply the /PI switch to the .TSK file specification.

The POSITION\_INDEPENDENT argument is identical to the PIC argument.

If you want to use more than one argument, enclose them in parentheses, separated by commas.

## LINK (Cont.)

**/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 directory, 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 do not give a file specification, the default is LB0:[1,1]ODT.OBJ, which is the DIGITAL-supplied utility, ODT (On-line Debugging Tool). ODT is incorporated in the task that 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 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.

In TKB format, apply the /DA switch to the .TSK file specification if you want 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[:taskspec]**

Synonym for the /TASK qualifier. See the /TASK qualifier discussion for more information.

**LINK (Cont.)****/[NO]EXTERNAL**

Specifies that the task be built to run with its header out of pool. The default is /EXTERNAL.

In TKB format, apply the /XH switch to the .TSK file specification.

**/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.

The following lists several restrictions that apply to tasks 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 an .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:FPP
/DEBUG[:filespec]
/LIBRARY
```

- The Task Builder is limited to the following options:

```
ASG
STACK
TASK
UNITS
EXTSCT
ACTFIL
MAXBUF
LIBR
COMMON
RESLIB
SUPLIB
RESSUP
```

In addition, the map format is nonstandard and undocumented.

## LINK (Cont.)

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 co-tree (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.

## /[NO]HEADER

Specifies that a header should be excluded from the task image. If you are building a shared region or a driver, you should specify this qualifier. If you use this qualifier, you cannot use the STACK option. The default is the /HEADER qualifier.

In TKB format, apply the /-HD switch to either the .TSK or .STB file specification. See the /SHAREABLE qualifier discussion.

## /[NO]IO\_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 the /IO\_PAGE qualifier.

## /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 unless you want to supply a name for the map file. If you supply a name, the map file appears in your current directory, but is not printed on the line printer.

In TKB format, apply the /-SH switch to the .MAP file specification.

**LINK (Cont.)****/MAP[:filespec]**

Specifies that you want a memory allocation, or map, file produced. If you use /MAP as a command qualifier without a filespec argument, the map appears in your directory with a file type of .MAP and a file name derived from the name of the first input file named in the command line. The map is also spooled to the line printer.

If you use /MAP with a filespec argument, either as a command qualifier or a file specification qualifier, the map appears in your directory 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 filespec argument, the map appears in your directory 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 the following 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]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. The argument n specifies the highest physical address of a task on the target system and can be 28 (the default) or 30.

The default is the /MEMORY\_MANAGEMENT qualifier.

In TKB format, apply the /-MM switch to the .TSK file specification.



## LINK (Cont.)

/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 6-1. See Section 6.5 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 you can stop a task build in this way, you must include the /OPTIONS qualifier in the LINK command line. This assures the availability of the ABORT option. If you are typing the command line and you make a mistake, specify the /OPTIONS qualifier, press RETURN, and enter the ABORT=0 option. Then press RETURN and the task build will abort.

/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. The /OVERLAY\_DESCRIPTION qualifier 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.

## LINK (Cont.)

### /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. The /RECEIVE qualifier 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 the /RESIDENT\_OVERLAY qualifier.

In TKB format, apply the /RO switch to the .TSK file specification.

### /SAVE

The /SAVE qualifier specifies that DCL retains the indirect command file created by DCL to pass your LINK command to TKB. See Section 6.1.1 for a description of DCL's handling of the LINK command. If you include the /SAVE qualifier 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

## LINK (Cont.)

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.

## /[NO]SEGREGATE

Causes the Task Builder to order program sections alphabetically by name within access code (RO followed by RW). If you also specify the /SEQUENTIAL qualifier, TKB orders program sections in their input order by access code.

The /NOSEGREGATE qualifier is the default. TKB interleaves RO and RW program sections. When combined with the /SEQUENTIAL qualifier, the /NOSEGREGATE qualifier results in a task with program sections allocated in input order with its RW and RO sections interleaved. If you use the /NOSEQUENTIAL and /NOSEGREGATE qualifiers 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. Generally, 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 the /SEGREGATE qualifier discussion.

In TKB format, apply the /SQ switch to the .TSK file specification.

```
/SHAREABLE[:arg]
    TASK
    COMMON
    LIBRARY
```

## LINK (Cont.)

Informs the Task Builder that one of three things is happening depending on the argument applied to this qualifier:

1. The /SHAREABLE:TASK qualifier identifies the task as a multiuser task. Such tasks permit more than one user to share the read-only portion of a single task. When you specify the /SHAREABLE:TASK qualifier, the Task Builder divides the task into two regions: region 0 contains the read-write portion of the task and region 1 contains the read-only portion of the task. When multiuser tasks are installed, multiple requests for the task to run cause the system to duplicate only the read-write portion of the task for each request after the first. The ROPAR option permits you to name the portion in which region 1 is to reside. In TKB format, apply the /MU switch to the .TSK file specification.

The /SHAREABLE:TASK qualifier defaults to the /SEGREGATE qualifier.

2. The /SHAREABLE:COMMON qualifier informs the Task Builder that a shareable common is being built. You should always use the /NOHEADER qualifier with the /SHAREABLE:COMMON qualifier. 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 do 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 with the /SHAREABLE qualifier contains all defined program sections. In TKB format, apply the /CO switch to the .TSK or .STB file specification. The /SHAREABLE:TASK qualifier defaults to the /SEGREGATE qualifier.
3. The /SHAREABLE:LIBRARY qualifier informs the Task Builder that a shareable library is being built. You should always use the /NOHEADER qualifier with the /SHAREABLE:LIBRARY qualifier. 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 do 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 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.

## LINK (Cont.)

**/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.

**/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 filespec 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 directory, but is not spooled to the line printer. See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information.

## LINK (Cont.)

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.

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.

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 as each instruction is completed during the running of the task.

In TKB format, apply the /TR switch to the .TSK file specification.

### /[NO]WARNINGS

Suppresses diagnostic messages issued by the Task Builder. The following two messages suppressed are:

n undefined symbols segment "segname"

and

Module "modulename" multiply defines P-section "psectname"

The default is the /WARNINGS qualifier.

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 the /WIDE qualifier.

If you include this qualifier, you include 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 directory, but is not spooled to the line printer.

In TKB format, apply the /WI switch to the .MAP file specification.

**Parameter Qualifier Description**

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. The /NOCONCATENATE qualifier 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.

**/INCLUDE:module1[:...modulen]**

You can specify 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 the /LIBRARY and the /INCLUDE qualifiers.

**/[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 the /GLOBALS qualifier.

In TKB format, apply the /-MA switch to the input file.

**/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 the /INCLUDE qualifier discussion.

**/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 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.



## LINK (Cont.)

Table 6-1  
Task Builder Options

Description and Language Key <sup>1</sup>	Format and Defaults <sup>2</sup>
<b>Control</b>	
Direct TKB to terminate build (M,H)	ABORT=0 Number used for format only. Any number will do.
<b>Identification</b>	
Declare task name (M,H)	TASK=1-6 alphanumeric characters  TASK=first six characters of task image file name
Set task's User	UIC=[g,m]
Identification Code (M,H)	UIC=terminal UIC
Declare task's priority	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 the partition in which the read-only portion of a multiuser task is to reside (M,H)	ROPAR=parname ROPAR=GEN
Declare the completion routine the supervisor mode library will use to return control to your program in user mode (M,H)	CMPRT=1-6 alphanumeric characters No default
Declare number of open active files (H)	ACTFIL=decimal integer ACTFIL=4

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/P-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
<b>Identification</b>	
Declare extension to FORTRAN record buffer (H)	MAXBUF=decimal integer for record buffer size in bytes (larger than default) MAXBUF=132.
Declare extension of buffer used for process- ing 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	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 addi- tional window blocks needed by task (M,H)	WNDWS=1-23 WNDWS=0
Declare size of stack	STACK=decimal integer for number of words needed for stack STACK=256.
<b>Sharing Storage</b>	
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	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
<b>Sharing Storage</b>	
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
Declare intention to access a user-owned supervisor mode library	RESSUP=filespec/[-]SV[: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
Declare intention to access a system-owned supervisor mode library	SUPLIB=filespec/[-]SV[:apr] Standard defaults on filespec
<b>Device</b>	
Declare number of I/O 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
<b>Alterations</b>	
Declare definition of global symbol (M)	GBLDEF=symbol-name:symbol-value No default
Declare global symbol reference (M)	GBLREF=symbol-name No default

(continued on next page)

## LINK (Cont.)

Table 6-1 (Cont.)  
Task Builder Options

Description and Language Key	Format and Defaults
	Alterations
Declare series of patch values; patches I-space in I- and D-space tasks (M)	ABSPAT=seg-name:address:val-1 . . . (up to 8) No default
Declares series of patch values; patches D-space of I- and D-space tasks	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	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

## 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 dollar sign 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 WRAY.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.

```
$ 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 6.1.1.) The entire LINK command is entered by the final RETURN in response to the Option? prompt.

```
$ LINK/OPTIONS/NOMEMORYMANWRAY (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)
```

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 WRAY, 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 for the /SAVED qualifier.

**LINK (Cont.)****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."

Note that nonprivileged users can build privileged tasks and high-priority tasks. However, those users cannot run 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

### 6.6 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 through LINK. The Task Builder (TKB) searches, in the following order, for definitions of all global symbols referenced in your program:

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 (that you specify) for macros listed in .MCALL statements and called in your source program, before searching the system macro library. See the RSX-11M/M-PLUS Guide to Program Development for more information on MACRO-11.

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 that you want to have rapid access to, 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.

#### 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 want to use indirect command files with the LIBRARY command, the indirect command file must be in LBR format, not DCL format. Only one level of indirect referencing is permitted.

You should be familiar with the way the MACRO-11 Assembler and the Task Builder use libraries. For more information, 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.

You cannot use wildcards (the percent sign (%) and the asterisk (\*)) 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:

1. Object module libraries. These can be user libraries or one of the following DIGITAL-supplied libraries:

LB0:[001001]SYSLIB.OLB	System library (this is the library that is searched by the Task Builder by default)
LB0:[001001]VMLLIB.OLB	Virtual Memory Management library
LB0:[001001]EXELIB.OLB	Executive library
LB0:[001001]RMSLIB.OLB	Record Management Services library
LB0:[001001]FOROTS.OLB	FORTTRAN-IV library (optional)
LB0:[001001]F4POTS.OLB	FORTTRAN-IV-PLUS library (optional)

**LIBRARY (Cont.)**

2. Macro libraries. These can be user libraries or one of the following DIGITAL-supplied libraries:

LB0:[001001]RSXMAC.SML	System macro library. This is the library searched by the assembler by default. (Note nonstandard file type.)
LB0:[001001]EXEMAC.MLB	Executive macro library
LB0:[001001]RMSMAC.MLB	Record Management Services library

3. 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-PLUS systems. Specifically, the LIBRARY command has no relationship to supervisor mode libraries or resident libraries created by the Task Builder.

On RSX-11M-PLUS, there are also two directories called libraries that have no relationship to the LIBRARY command: the system directory LB0:[1,54], which contains privileged system tasks (such as commands), and the library directory LB0:[3,54], which contains nonprivileged system tasks (such as the utilities).

All error messages are described in Section 6.6.9. Some error messages are closely related to particular LIBRARY operations. Descriptions of these messages are repeated in the section which discusses the corresponding operation.

**6.6.1 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**

```
GLOBALS:n  
MODULES:n  
BLOCKS:n
```

**LIBRARY (Cont.)****Parameters****libspec**

Specifies the name of the library file to be compressed. The default type is .OLB, specifying an object module library. If you want 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**

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. This parameter is optional.

**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 value for n is the number of global symbols allocated in the old library. The maximum value for n is 4096. The value of n is always forced to 0 for macro and universal libraries.

**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: [001001]SYSLIB.OLB (RET)
```

This example compresses the system object module library.

**Note**

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.

## LIBRARY (Cont.)

### 6.6.2 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
```

#### 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 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. The value of n can range from 0 through 4096.

**LIBRARY (Cont.)****MODULES:n**

Specifies the number of entries to allocate in the module name table. The default value is 256. The value of n can range 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 want to use the same global symbols in more than one module. /GLOBALS is the default.

**/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.

**/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.

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.

This /SQUEEZE qualifier is the equivalent of the LBR /SZ switch applied to the output library when using the /CR switch.

**/UNIVERSAL**

Specifies that the library being created is a universal library.

## LIBRARY (Cont.)

### 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 then 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.

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 want to preserve. The following line of code moves the ASCII value of the semicolon to register 1:

```
MOVB #' ;,R1
```

However, the code emerges after the /SQUEEZE as the following:

```
MOVB    #'
```

To preserve the meaningful semicolon, use the following form:

```
MOVB    #' ;,R1 ;THIS DUMMY COMMENT SAVES THE CODE
```

which emerges after the /SQUEEZE as the following:

```
MOVB    #' ;,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
```

Keep this convention in mind whenever you are using the /SQUEEZE qualifier.

## LIBRARY (Cont.)

## 6.6.3 LIBRARY/DELETE

LIBRARY/DELETE deletes modules from any library. See the LIBRARY/REMOVE qualifier discussion for information on 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]

Specifies the modules that are to be deleted. 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 6.6.1.

## LIBRARY (Cont.)

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 the LIBRARY/LIST qualifier, Section 6.6.6, to check which modules have been deleted and which have not.

See the LIBRARY/REMOVE qualifier discussion for information on removing global symbols from libraries.

### 6.6.4 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 want 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.



**LIBRARY (Cont.)****Command Qualifier**

/OUTPUT

Specifies the file to which the extracted modules or macros are to be written. If you specify the /OUTPUT qualifier 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 LBO:[001001]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.

```
$ LIBR/EXTRACT/OUT:TI: LBO:[001001]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 spools it to the terminal.

**Note**

This command is the equivalent of the LBR /EX switch applied to the library file as the input file.

LIBRARY/EXTRACT has no effect on the library from which the modules are extracted.

**6.6.5 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
```

## LIBRARY (Cont.)

### Parameters

#### libspec

Specifies the library into which modules are to be inserted.

The default file type is .OLB. If you want 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 and .OBJ for object module libraries, and .UNI for universal libraries.

### Command Qualifiers

#### /[NO]GLOBALS

Specifies whether entry points for the specified modules should be included in the entry point table.

Use the /NOGLOBALS qualifier if you want 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.

**LIBRARY (Cont.)**

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 6.2.2 for important information on the /SQUEEZE qualifier.

This command is the equivalent of the LBR /SZ switch applied to the output file.

**Example**

```
$ LIBRARY/INSERT (RET)
Library? LB0:[001001]SYSLIB (RET)
Module(s)? USERSUB (RET)
```

This example inserts object modules from the file USERSUB.OBJ into the system object module library, LB0:[001001]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 inserts 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.

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 the LIBRARY/LIST qualifier to check which modules have been inserted and which have not.

**LIBRARY (Cont.)****6.6.6 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 want listed. The default file type is .OLB. If you want 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 directory are the same as those of the library you are listing and not your current defaults. See the examples.

/BRIEF

Specifies that you want the list to include only the module names. This is the default. See the examples.

/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.

## LIBRARY (Cont.)

## Examples

```
$ LIBRARY/LIST LB0:[001001]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 LB0:[001001]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
```

```
.
.
.
```

This example shows the display from LIBRARY/LIS/FULL, which includes full module descriptions. This information appears on your terminal.

**LIBRARY (Cont.)**

```
$ LIBRARY/LIST/NAMES LB0:[001001]SYSLIB 
```

```
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
```

```
    ALT PR I
```

```
** Module:ANSPAD
```

```
    ..ANSP
```

```
** Module:ARITH
```

```
    $DIV      $MUL
```

```
    .
    .
    .
```

This example shows the display from LIBRARY/LIST/NAMES, which lists entry points for each module.

## LIBRARY (Cont.)

```
$ LIBRARY/LIST/FULL/NAMES LB0:[001001]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      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:20:0P6
```

```
..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)
DU2: [LAYLA] TT26:
$ LIBRARY/LIST:FRED.LIS LB0:[001001]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 directory on the default device. The error message indicates that the listing file cannot be opened. This is because LIBRARY attempted to place FRED.LIS on LB0: in [001001], where the user did not have file creation privileges. See the next example.

## LIBRARY (Cont.)

```
$ SHOW DEFAULT (RET)
DU2: [LAYLA]      TT26:
$ LIBRARY/LIST:DU2:[LAYLA]FRED.LIS LB0:[001001]SYSLIB (RET)
```

In this example, the user explicitly stated the device and directory for the listing file and the command succeeded.

### Note

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. (See Command Qualifiers.)

### 6.6.7 LIBRARY/REMOVE

LIBRARY/REMOVE removes global symbols from a library. See the LIBRARY/DELETE qualifier discussion 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 want to remove global symbols.

global[,s]

Specifies the global symbols you want 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.



**LIBRARY (Cont.)****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 the LIBRARY/LIST qualifier to check which globals have been deleted and which have not.

**6.6.8 LIBRARY/REPLACE**

The LIBRARY/REPLACE qualifier 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 want 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 the example. If the module does not exist, it is inserted, but no message is printed.

## LIBRARY (Cont.)

### Command Qualifiers

`/[NO]GLOBALS`

Specifies whether (or not) entry points for the specified modules are to be included in the entry point table. The default is the `/GLOBALS` qualifier.

`/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.

`/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 6.6.2 for more information on the `/SQUEEZE` qualifier.

### 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

There must be enough space in the library's tables for both the modules being replaced and their replacements, because the new modules are entered before the old modules are deleted.

The old modules are logically deleted only, that is, all references to those modules are removed, but not the module itself. You must use `LIBRARY/COMPRESS` to eliminate the modules and free the space they occupy.

**LIBRARY (Cont.)****6.6.9 Error Messages from LIBRARY**

LIBRARY has two kinds of error messages, diagnostic and fatal.

Diagnostic messages are marked \*DIAG\*. They inform you of conditions that require your consideration but which 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 LIBRARY 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 can result from corrupted library files.

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 is either not a library or 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 the RSX-11M/M-PLUS Utilities Manual.

LBR---Command I/O error

**Explanation:** There are two possible causes:

1. The problem may be with the physical device, which, for example, may not be spun up.
2. The file may be corrupted or have an incorrect format, where, for example, a record length exceeds 132 bytes.

**User Action:** Determine the condition that caused the message and correct the condition. Reenter the command line.

LBR---Command syntax error

**Explanation:** The reprinted command line has a syntax error.

**User Action:** Check for proper syntax and reenter the command line.

## LIBRARY (Cont.)

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.

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.

## LIBRARY (Cont.)

### 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. LIBRARY 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.

### 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 LIBRARY is running is too small.

**User Action:** See your system manager.

### 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 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.

## LIBRARY (Cont.)

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 with the physical device, which, for example, may not be spun up.
2. The input file is corrupted or in the wrong format, where, for example, a record exceeds 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.

LBR---Mark for delete failure on LIBRARY work file

**Explanation:** LIBRARY 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 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.

## LIBRARY (Cont.)

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 seven following conditions:

1. The user directory area is protected against a Write operation.
2. There is a problem with the physical device, which, for example, may not be spun up.
3. The volume is not mounted.
4. The UFD 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.

LBR---Open failure on LIBRARY work file

**Explanation:** The file system detected an error while attempting to open the LIBRARY work file. The LIBRARY work file is created on the volume from which LIBRARY was installed. One of the following four 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.

## LIBRARY (Cont.)

### LBR---Output error on filename

**Explanation:** A write error has occurred on the output file. One of the following three 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.

### LBR---Work file I/O error

**Explanation:** A write error has occurred on the LIBRARY work file. One of the following three 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.



## SET GROUPFLAGS

### 6.7 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 group 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.

#### 6.7.1 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 group 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 want 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 login group. Privileged users can create and delete group global event flags without restriction.

/DELETE

Specifies that you want to delete a set of group global event flags. The argument n is the group with which the flags are associated.

## SET GROUPFLAGS (Cont.)

### 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 is 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

### 6.7.2 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 flags 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-PLUS 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 7.14.3.2.

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 also install tasks through the RUN command. However, these tasks remain installed after execution.

Many system tasks can be executed 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 do not necessarily reside in memory or compete for system resources. Most installed tasks are dormant, meaning they are installed but have not been requested to run. If a dormant task resides in memory, the task requires 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 the following 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 the following two substates:

Ready-to-run	Competing for CPU time on the basis of priority
--------------	---

## RUNNING TASKS

Blocked            Unable to compete for CPU time because a needed resource is not available, a synchronization factor is present, or because the STOP/BLOCK command is in effect (Section 7.6)

You can display the list of active tasks with SHOW TASKS/ACTIVE. See Section 7.14.3.1.

Tasks run at priorities of 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. These tasks can then be 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 that are 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 the 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.

Likewise, the following batch command, issued from virtual terminal VT4:

```
$ SHOW TIME
```

results in a task named SHOV4 in the STD while the task is executing.

## RUNNING TASKS

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.

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 want 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 because all task names in the STD are unique; that is, you cannot have two tasks named TT10 installed at the same time. ( See the examples in Section 7.2.1). Finally, most DCL commands work by

## RUNNING TASKS

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 in the form PIP10.

See Section 7.2 for information on including the dollar sign (\$) in a RUN command.

On RSX-11M-PLUS systems with more than 64 terminals, tasks run from the higher-numbered terminals cannot be named directly after their terminals because of restrictions on the number of characters in a task name. The following discussion describes the problem and its solution.

Remember that terminal numbers are octal. The first 64 terminals on an RSX-11M-PLUS system are numbered 1 through 77 (octal). The system supports as many as 256(decimal) terminals. Terminals 65 through 256 are numbered 100 through 377(octal). Tasks initiated from these terminals cannot be named directly after these terminals because their numbers, if used explicitly, are too large for task names (which are limited to six characters). For example, SHOT107 is an illegal task name. Therefore, when the RUN command is issued from terminals numbered 100 through 377(octal), a slightly different derivation of task names is employed. See the following example.

Assume your terminal is TT107:. The following command:

```
$ RUN ROBOT (RET)
```

initiates a task named TTA7. The letter A stands for the first two digits of the unit number of the terminal, 10 in this case. To minimize the confusion resulting from this convention, the display from SHOW TASKS/ACTIVE includes the name of the initiating terminal in parentheses next to the name of the active task. The SHOW task itself will be named SHOTA7.

The numbering system replaces the first two digits of the unit number with a single letter. Here is a full list of the possible task numbers and the equivalent terminal unit numbers:

Task Number	Terminal Unit Number
0-77	0-77
AO-A7	100-107
BO-B7	110-117
CO-C7	120-127
DO-D7	130-137
EO-E7	140-147
FO-F7	150-157
GO-G7	160-167
HO-H7	170-177
10-17	200-207

(continued on next page)



## RUNNING TASKS

Task Number	Terminal Unit Number
J0-J7	210-217
K0-K7	220-227
L0-L7	230-237
M0-M7	240-247
N0-N7	250-257
O0-O7	260-267
P0-P7	270-277
Q0-Q7	300-307
R0-R7	310-317
S0-S7	320-327
T0-T7	330-337
U0-U7	340-347
V0-V7	350-357
W0-W7	360-367
X0-X7	370-377

Only uninstalled tasks initiated with the RUN command have task names preceded by VT, TT, or HT (or RT). Tasks initiated by other DCL commands are named after the first three letters of the command that initiated the task, plus a T, V, or H (or R) (for physical, virtual, or DECnet host terminal) plus the unit number.

### 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 directory. 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 directory, or library directory.
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 directory on the default device for a task image file named PUPPET.TSK. When the file is found, the image is read into 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 DU0:MUPPET (RET)
```

## RUNNING TASKS

directs the operating system to do the following:

1. Look in the default directory on device DU0: for a file named MUPPET.TSK.
2. Install and run the task image.
3. 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 the system and library directories. The dollar sign (\$) in a file specification indicates that you want to use this form of RUN. Using this convention relieves you of the necessity of knowing exactly where on the system various system tasks are stored. 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 one of these directories, 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 for more information on RMS-11 tasks and utilities. (Your system manager may also have added other task image files not supplied by DIGITAL to these directories for your convenience.)

Copies of many or all privileged 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. In addition, copies of many or all nonprivileged system tasks are placed in the library directory on pseudo device LB:. The directory is usually [3,54], but it too can be some other number. You can display the identity of this directory with the command SHOW LIBRARY/DIRECTORY. You can also list the contents of the directory with a DIRECTORY command.

The system responds to the dollar sign (\$) in a file specification as follows. The command:

```
$ RUN $TKB (RET)
```

directs the operating system to do the following:

1. Look for a file called TKB.TSK.
2. Install and run the task image.
3. Remove the task image when it has completed execution.

## RUNNING TASKS

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.

If the command had named a nonprivileged task, such as in the following command

```
$ RUN $QIX (RET)
```

the system would still have started by searching the system directory, but, failing to find QIX.TSK there, would have gone on to the library directory to search for QIX.TSK there.

### NOTE

Privileged users can also use the dollar sign (\$) with the INSTALL command. See Section 7.8 for more information.

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 is not installed by the system (because it is already installed) nor is it removed when it has completed its execution. The task runs under the name TURTLE, not TT10.

If the system does not find an installed task of the same name you give, the command defaults to the install-run-remove form and looks in the default directory on the default device for a file named TURTLE.TSK. The system then installs, runs, and removes the task, as described earlier.

Finally, privileged users can use RUN to schedule the execution of a task in several forms. Scheduling options include the following:

- 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 beginning of the next interval, that is, the next hour, minute, second, or tick.
- 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.

## RUNNING TASKS

### 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  
/READ PARTITION:pname  
/[NO]SLAVE  
/STATUS:arg  
        TASK  
        COMMAND  
/TASK_NAME:taskname  
/TIME_LIMIT:n[u]  
/UIC:[g,m]
```

#### 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 and then in the library 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, allowing you to run checkpointable tasks as noncheckpointable, or the converse.

## RUNNING TASKS

/COMMAND:"taskcommand"

Passes a command to the task you are running. The command must be inside the quotation marks (") and not more than 40 characters long. For example, the following command:

```
$ RUN/COMMAND: "/LI" MAMBO 
```

runs MAMBO and then passes the /LI switch to it.

/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.

/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. For more information on Postmortem Dumps see the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual .

/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) .

/READ\_PARTITION:pname

Installs the read-only portion of a multiuser task into a specified partition (pname). If the specified partition does not exist, the read-only segment is installed in the same partition as the task.

## RUNNING TASKS

`/[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:TASK` and need not be specified.

The `/STATUS:TASK` qualifier may be necessary in user batch jobs or in indirect command files that run installed tasks, where the batch processor must wait for the installed task to exit before attempting to execute the next command.

`/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 want 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 (\$).

`/TIME_LIMIT:n[u]`

Allows you to limit the amount of CPU time the task can run. The default is `/TIME_LIMIT:3M`. The "M" stands for minutes. If you give simply a number as an argument, the time unit defaults to minutes, but you can also specify a time limit in seconds by using a number and an "S", as in `/TIME_LIMIT:30S`.

`/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, assuming there is no installed task named ROBOT.

```
$ 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 file type specified, will run the task from the directory.

The following examples illustrate the task-naming convention. These examples assume that a SET TERMINAL/NOSERIAL command has been issued to allow more than one task to run at a time. See Chapter 3.

The terminal number in parentheses indicates the terminal from which the tasks are being run.

```
$ RUN ROBOT (RET)
$ SHOW TASK/ACTIVE (RET)
MCR... (TT107:)
SHOTA7 (TT107:)
TTA7 (TT107:)
```

This example illustrates the task-naming convention as applied from a terminal with a unit number greater than 77(octal), in this case, TT107:. Notice the name given to SHOW task.

In the following example, the terminal number is less than 77 (octal).

```
$ RUN/TASK_NAME:CAPEK ROBOT (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR... (TT10:)
SHOT10 (TT10:)
CAPEK (TT10:)
```

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 ROBOT (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR... (TT10:)
SHOT10 (TT10:)
TT10 (TT10:)
$ RUN ASIMOV (RET)
RUN -- Task name already in use
$ RUN/TASK_NAME:BINDERASIMOV (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR... (TT10:)
SHOT10: (TT10:)
TT10 (TT10:)
BINDER (TT10:)
```

This example shows how you can run two tasks simultaneously using the RUN command and the /TASK NAME qualifier.

## RUNNING TASKS

### Notes

If you are in serial mode, the return of the dollar sign prompt (\$) signifies that the task initiated by the RUN command has exited. If you are not in serial mode, the task could still be executing. However, you need not wait for the prompt to return to issue another command. Simply enter a carriage 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 want to run, the RUN command will run the installed task rather than the one you want to run. In such an instance, you must include the file type or some part of a file specification, including a dollar sign (\$), to force the RUN command to access a 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.

### 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]
```



## RUNNING TASKS

### 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 7.14.5.

#### /DELAY:nu

Specifies that the task be run after the stated amount of time passes. This qualifier is privileged. The argument nu specifies the delay as a number of units of time, where n is the number of units, and u is one of the following units:

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 time 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
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, where n is the number of units, and u is one of the following time units:

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

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

## RUNNING TASKS

/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 user batch jobs or in indirect command files that run installed tasks where the batch processor must wait for the installed task to exit before attempting to execute the next command. See the examples using /STATUS:TASK.

This qualifier can be only specified separately or with the /UIC:[g,m] 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 one of the following time units:

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

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

/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 5 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 10 minutes, and every hour thereafter.

## RUNNING TASKS

```
$ 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.

```
.  
.  
.  
$INSTALL $HIFI  
$RUN/STATUS: TASK HIFI  
$EOJ
```

This example illustrates the use of the /STATUS:TASK qualifier in a user batch job. First, the job installs the task HIFI from the library directory, then the task is processed. If the user had not included the /STATUS:TASK qualifier, the RUN command would have returned a status of success, the batch processor would have executed the EOJ, and then, the EOJ would have aborted the task that had just started running. With the qualifier, the EOJ is not executed until after the task has exited. See the next example.

```
.  
.  
.  
$INSTALL CAVIAR  
$RUN/STATUS:TASK CAVIAR  
$PRINT/FLAG PAGE FISHEGGS.LST, SHADROE.LST  
.  
.  
.
```

This example also illustrates the use of the /STATUS:TASK qualifier in a user batch job. First, the job installs the task CAVIAR. When processed, this task writes two listing files: FISHEGGS.LST and SHADROE.LST. After these files have been written and closed, the batch job prints them both. With the /STATUS:TASK qualifier on the RUN command, the batch processor waits for the installed task CAVIAR to exit before attempting to print the files. Without the qualifier, the batch processor would attempt to print the files while they were still being written.

## RUNNING TASKS

### Notes

The dollar sign prompt (\$) returns immediately after you issue a RUN command for an installed task and does not indicate that the task has completed its execution. If you specified the /STATUS:TASK qualifier, the dollar sign prompt signifies that the task initiated by the RUN command has exited.

You can display entries in the clock queue with SHOW CLOCK\_QUEUE, Section 7.14.5.

The clock queue is kept in pool and must be restored whenever the system is rebootsrapped.

### 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  
/[NO]POSTMORTEM  
/TASK  
/TERMINAL:ttnn:

### 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 want 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 want 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 the error message, "Illegal task name."

### Command Qualifiers

#### /COMMAND

Specifies that you want to abort a command. This is the default qualifier and need not be specified.

## ABORT (Cont.)

/[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.

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.

/TERMINAL:ttnn:

Specifies that a task from some terminal other than your own be aborted. This is a privileged qualifier.

### 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 the 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 then, RUN was specified. The task was aborted and removed, and the message displayed. If your RSX-11M-PLUS system supports CTRL/C ABORT, issuing CTRL/C alone would have been sufficient. See Chapter 3, for more information.

```
$ 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 the next example.

## ABORT (Cont.)

```
$ DIRECTORY *.RNO (RET)

Directory DU2:[303,5]
20-MAY-85 16:28

ALTCLI.RNO;3          213.          27-APR-85 16:52
ANNI.RNO;6            3.            30-MAR-85 09:30
APRCOM. RNO; 4         9.            15-APR-85 10:35
BEST.RNO;5            40.           28-APR-85 15:50
CATCH.RNO;4           7.            22-APR-85 09:39
CHAP11SMG.RNO;13      49.            13-MAY-85 16:36
CHAP12SMG.RNO;51     223.            08-APR-85 13:15
CHARSET.RNO;5         6.            15-MAY-85 11:57 (CTRL/C)
DCL>ABORT DIRECTORY (RET)
16:29:45 Task "DIRT6" terminated
        Aborted via directive or CLI
```

In this example, the user issued a DIRECTORY command and soon saw that the file being checked for was present. Rather than wait for the directory listing to complete, the user typed CTRL/C, and then, entered the ABORT command in response to the explicit DCL prompt. The ABORT command defaulted to ABORT/COMMAND. If your system supports CTRL/C ABORT, the CTRL/C alone would have been sufficient.

```
$ ABORT/TERMINAL:TT3: MACRO (RET)
```

This example, issued by a privileged user, aborts the MACRO task running on another terminal. Notification of the abort appears on TT3:, but not on the terminal from which the ABORT command was issued.

```
$ 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.

## ABORT (Cont.)

### Notes

A is the short form of ABORT. See also the discussion of CTRL/C in Chapter 3, under SET TERMINAL.

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.
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:. The TKTN routine can list a number of messages about the abort. These messages are explained in Chapter 10.
4. Releases any memory allocated to the task if the task is not fixed.

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.

### 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 the command for the 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 [/TERMINAL:ttnn:] taskname]
```

### Command Qualifier

```
/TERMINAL:ddnn:
```

### 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.

### Command Qualifier

```
/TERMINAL:ttnn:
```

Allows you to continue a suspended task running on some other terminal. This is a privileged qualifier.

### 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
BEGS 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

## CONTINUE (Cont.)

under STATUS on the second line, the mnemonic SPN verifies that the task was suspended. The user issued CONTINUE without naming the task. Because 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.

Batch jobs may include the noninteractive form of CONTINUE as a no-operation command. See the RSX-11M/M-PLUS Batch and Queue Operations Manual for more information.

### 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 want to eliminate.

### Example

```
$ RUN/DELAY:5M ARDVRK (RET)
$ SHOW CLOCK_QUEUE (ET)
ARDVRK Scheduled at 17-MAY-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. First, 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. Finally, 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 7.14.5 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[/qualifier] [taskname]

Command Qualifier

/TERMINAL:ttnn:

### 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.

### Command Qualifier

/TERMINAL:ttnn:

Specifies that a task running from the named terminal is to be blocked. This is a privileged qualifier.

### Example

```
$ RUN ADVENT (RET)
$ SHO TASK/ACTIVE (RET)
MCR...
SHOTS
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, first, the user on terminal TT5: ran a task with the RUN command, giving the task the name TT5 by default. Then, the user issued the STOP/BLOCK command, without naming the task. The /FULL

## STOP/BLOCK (Cont.)

display from SHOW TASKS shows task TT5's status as blocked (BLK flag). Finally, 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 (or CTRL/C, if enabled) are the only means of canceling the STOP/BLOCK command.

A blocked task has an effective priority of 0. If the task is set 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.

A task can also be blocked by the system while waiting for some needed resource or for synchronization.

There are no prompts from this command.

See Section 7.14.3.1 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 for more information.

## 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 want 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 RSX-11M/M-PLUS Batch and Queue Operations Manual and the RSX-11M/M-PLUS System Management Guide for more information. For information on STOP\$\$, see the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual.

## 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[/qualifier] [taskname]

#### Command Qualifier

/TERMINAL:ttnn:

#### 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.

#### Command Qualifier

/TERMINAL:ttnn:

Specifies that a task running from the named terminal is to be unblocked. This is a privileged qualifier.

#### Notes

See the description of STOP/BLOCK for an example.

START/UNBLOCK does not counteract the Executive blocking tasks 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]RESIDENT_HEADER
/[NO]SLAVE
/TASK_NAME:taskname
/TRANSLATION_ROUTINE:n
/UIC:[g,m]
/[NO]WRITEBACK
```

### Parameter

[ $\$$ ]filespec

Specifies the name of the task image file containing the task you want to install. The default file type is .TSK. If present, the dollar sign ( $\$$ ) directs the system to search for the file in the system directory and in the library 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.



## INSTALL (Cont.)

### 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.

/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" SAMBA (RET)
```

installs SAMBA and then passes the /LI switch to it.

/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 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 a synchronous system trap (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 range from 0 through 250 (decimal).

## INSTALL (Cont.)

### `/READONLY_COMMON`

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

### `/[NO]RESIDENT_HEADER`

Specifies whether or not the Executive's copy of the task's header will be stored in the system's dynamic storage region (pool).

When you specify the `/RESIDENT_HEADER` qualifier, the task is installed as a resident header. When you specify the `/NORESIDENT_HEADER` qualifier (the default), the task is installed with an external header.

If you take the default (`/NORESIDENT_HEADER`), the task's header is not kept in pool. Instead, a copy of the header is placed in a reserved area immediately below the task image. The Executive uses the copy of the header rather than the actual header. When the task is checkpointed, the system writes the entire task image and the copy of the header into the checkpoint file. The header of the task is left unchanged.

### `/[NO]SLAVE`

Specifies whether or not the task is to be slaved.

Data set 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.

### `/TRANSLATION_ROUTINE:n`

Loads character translation routines so that the terminal driver can translate between different character sets. Character translation in the terminal driver allows terminals that conform to other standards to use the DIGITAL multinational character set. This qualifier is privileged.

INSTALL/TRANSLATION\_ROUTINE attaches ancillary control drivers (ACD) to the terminal driver. Each ACD handles a specific character translation routine. By using the character translation routines, you can select the translation to be performed between the character set used by the system and the

## INSTALL (Cont.)

one used by your terminal. Also, character translation allows the system and application programs to recognize the same character set, regardless of the type of terminal you are using. Finally, character translation is transparent and is performed for all characters independent of the type of I/O being used.

The full format for this qualifier is as follows:

INSTALL/TRANSLATION\_ROUTINE:n filespec [logical]

This command loads the character translation routine contained in the specified file and assigns an identifying number or name, or both, to the routine.

The n argument identifies the ACD that handles the translation routine in the specified file. The value that you specify for n overrides the internal number that is built into the ACD. The value for n can be either numeric or alphanumeric. If numeric, n must be an even number; it is a decimal number by default. If alphanumeric, n specifies a logical name that is equivalent to the ACD.

The file specification parameter specifies the file name that contains the translation routine you want to install. The DIGITAL-supplied translation routines are located in the system directory and have the following file names:

MCSVTW.TSK	For VT100-Wx model terminals with U.S. keyboard
MCSVT.TSK	For VT100 non-W model terminals that have the DIGITAL multinational character set on an add-on ROM
MCSVTWG.TSK	For VT100-WK/WJ model terminals with German keyboard
MCSVTWF.TSK	For VT100-WF/WE model terminals with French keyboard

The optional logical parameter assigns a logical name to the ACD that handles the specified translation routine.

To use the installed character translation routines, see the SET TERMINAL/TRANSLATION\_ROUTINE command description in Chapter 3.

/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.

## INSTALL (Cont.)

/[NO] WRITEBACK

Specifies whether or not a read/write common is to be written back to the original task image file on a checkpoint or remove operation. Taking the default (/NOWRITEBACK) means that the read/write common is to be checkpointed to the system checkpoint file.

### 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 then, 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]) and in the library directory (by convention, [3,54]) for a task image file named LATE.TSK, and then, 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 then, installs it as BUCKIN in the STD, assuming no other name was specified at link time. Task names are limited to six Radix-50 characters.

\$ **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 then, installs it as DOOR in the STD, overriding any task name specified at link time.

\$ **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 then, 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.

## INSTALL (Cont.)

Several situations require tasks to be explicitly installed through INSTALL. If you want other tasks to be able to communicate with the task while it is not active, it must be installed. If you want 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 want to sacrifice pool space and install the task to improve system performance.

You can display the attributes of installed tasks with SHOW TASKS/INSTALLED. See Section 7.14.3.2.

For information on Executive directives, see the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual.

### 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.

ACD---ACD already exists

**Explanation:** Command tried to install an ACD that is currently installed. The Ancillary Control Driver (ACD) task returns the message.

**User Action:** None.

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.

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.

## INSTALL (Cont.)

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 the SET DEVICE/CHECKPOINT command and reenter the command line.

INS---Checkpoint space too small, using checkpoint file

**Explanation:** This is a 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 then, reinstall it with a different value for /EXTENSION if you want.

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:** First, make sure you have specified the correct file, and then 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:** First, make sure you have specified the correct file, and then reenter the command.

## INSTALL (Cont.)

### INS---Illegal device ddnn:

**Explanation:** This is a 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 want.

**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:** First, make sure you specified the proper device, and then reenter the 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:** This is a 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. Because the system does not support dynamic allocation of checkpoint space, the system considers the task to be not checkpointable.

**User Action:** 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 a description of the /PR keyword (/PRIVILEGED qualifier to LINK).

INS---R/O partition parname not in system defaulting to task's partition

**Explanation:** This is a warning message. The command attempted to install a read-only common in a nonexistent partition. The common was installed in task's partition.

**User Action:** If you want, you can create the missing partition.



## INSTALL (Cont.)

INS---Specified partition too small

**Explanation:** The command attempted to install task in a partition that was smaller than the task.

**User Action:** First, increase the size of the partition, and then reenter the command.

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 the 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:** The 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.

## INSTALL (Cont.)

INS---Too many LUNs

**Explanation:** The task requested more than 250 LUNs to be assigned.

**User Action:** Recheck task build and code for source of the error, and correct it.

INS---WARNING---Privileged task overmaps the I/O page

**Explanation:** A privileged task that is mapped into the Executive is usually mapped into the I/O page as well, 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[/qualifier]
Task?      taskname

REMOVE [/qualifier] taskname

Command Qualifier

/REGION
/TRANSLATION ROUTINE:n
```

### Parameter

taskname

Specifies the name of the task you want to remove.

If you want to remove a region, specify the /REGION qualifier and the name of the region.

### Command Qualifier

/REGION

Specifies that you want to remove a region from the Common Block Directory.

/TRANSLATION\_ROUTINE:n

Removes the ancillary control driver (ACD) specified by the argument n. If you assigned a logical name for the ACD when you installed it, the name is deleted when you remove the ACD. This qualifier is privileged.

### Examples

```
$ REMOVE WONTON (RET)
```

This example removes the task named WONTON from the System Task Directory. It is no longer installed.

```
$ REMOVE/ REGION DONJON (RET)
```

This example removes the common region named DONJON from the Common Block Directory.

## REMOVE (Cont.)

### Notes

To remove an active task, you must first abort it.

A region cannot be removed if there are tasks installed that reference that region.

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 deattached all attached regions. See the RSX-11M/M-PLUS and Micro/R SX Executive Reference Manual for more information.

### Error Messages

ACD---ACD is busy but was marked for delete

**Explanation:** Command tried to remove an ACD that is still attached to a terminal. The ACD must be detached from all terminals before it can be removed. The Ancillary Control Driver task (ACD) returns the message.

**User Action:** None.

ACD---That ACD is not installed

**Explanation:** You tried to remove an ACD that is not currently installed. The Ancillary Control Driver task (ACD) returns the message.

**User Action:** Check the syntax and reenter the command.

# 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 [/qualifier]
Task?    taskname

FIX [/qualifier] taskname

Command Qualifiers

/READONLY SEGMENT
/REGION
```

### Parameter

taskname

Specifies the installed task or region you want to fix in memory.

Command Qualifiers

If you do not specify a qualifier, an installed task is fixed in memory.

/READONLY\_SEGMENT

Allows you to fix in memory the read-only segment of a multiuser task.

/REGION

Allows you to fix in memory a common region.

### 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
```

## FIX (Cont.)

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.

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, because 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 or region from memory. This allows tasks, which have been waiting for space in the partition where the fixed task resides, to compete for that space. This is a privileged command.

### Format

```
UNFIX[/qualifier]
Task? taskname

UNFIX[/qualifier] taskname
```

### Command Qualifiers

```
/REGION
/READONLY_SEGMENT
```

### Parameter

taskname

Specifies the name of the task you want to unfix. If you want to unfix a region, specify the name of the region for this parameter.

### Command Qualifiers

/REGION

Specifies that you want to unfix a common region rather than a task.

/READONLY\_SEGMENT

Specifies that you want to unfix the read-only segment of a multiuser task.

### Example

See the 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:** The command attempted to unfix a task that was not fixed.

**User Action:** None.

## ASSIGN/TASK

### 7.12 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 want to reassign.

ddnn:

Specifies the new device to which you want the LUN reassigned. This can be a physical device, pseudo device, or logical device name.

lun

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

#### Examples

```
$ ASSIGN/TASK:NIKNAK (RET)
Device? DU1: (RET)
Logical unit? 5 (RET)
```

This example overrides the assignment of LUN 5 established at link time and reassigns LUN 5 to DU1:.

```
$ 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:.



## ASSIGN/TASK (Cont.)

### 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. See Chapter 5, for more information.

### Error Message

ASS---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.

## DEBUG

### 7.13 DEBUG

DEBUG forces a task to trap to a debugger by setting the T-bit in the task's Processor Status Word. The task must have been built using the /DEBUG qualifier to the LINK command or it must have been issued the SVDB\$ Executive directive (Specify SST Vector Table for Debugging Aid)

DIGITAL offers two debuggers for use with this command:

1. ODT-11, the On-Line Debugging Tool
2. RSX-11M-PLUS FORTRAN-77 DEBUG

See the RSX-11M/M-PLUS and Micro/RSX Debugging Reference Manual for more information on ODT and the documentation accompanying the RSX-11M-PLUS FORTRAN-77 product (available separately) for more information on FORTRAN-77 DEBUG.

Nonprivileged users can issue this command for any nonprivileged task running from their terminal. Privileged users can name any task, but the command must be issued from the same terminal the task was run from.

#### Format

DEBUG [taskname]

#### Parameter

taskname

Specifies the name of the task you want to trap. If you do not specify a taskname, the default is a task named after the terminal from which the command was issued, such as TT5:.

#### Example

```
$ RUN JOE (RET)
$ SHOW TASKS/ACTIVE (RET)
MCR... (TT5:)
SHOT5 (TT5:)
TT5 (TT5:)
.
.
.
$ DEBUG (RET)
TE:002452
```

In this example, the task was built including ODT. The user issued the DEBUG command, which defaulted to the task named after the terminal. The display from ODT gives the T-bit exception code and the address following the location in error and issues its prompt. The user can then proceed with debugging.

## DEBUG (Cont.)

### Notes

The DEBUG command is useful when a task is caught in an infinite loop between breakpoints, or is waiting for an incorrect or nonexistent event flag.

See the RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual for information on the SVDB\$ directive.

See the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual for more information on the /DEBUG qualifier to LINK.

### Error Message

DEB---"time" "taskname" does not contain a debugging aid

**Explanation:** An attempt was made to debug a task that was not built with a debugging aid.

**User Action:** None. This cannot be done.

### 7.14 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

### 7.14.1 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-PLUS partitions can be of two types: system-controlled or device commons.

#### Format

SET PARTITION:parname/qualifier[s]

Base? n

Size? n

SET NOPARTITION:parname

#### Required Qualifiers

/BASE:n

/SIZE:n

#### Qualifier

/TOP:arg

#### Partition Types

/DEVICE

/DIAGNOSTIC

/SYSTEM

#### 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. The argument n is the physical base address specified in 64-byte units.

/SIZE:n

Specifies the size of the partition. The argument 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, the argument 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	nnn.K	(nnn.*2048.	0<base<2047.K	0.<size<32.K

For system-controlled partitions, the values in Table 7-1 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.

## Command Qualifier

/TOP:arg

Moves the top boundary of the specified system according to the form the argument is stated in. The argument n 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

/DEVICE

Specifies a device partition. 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.

## SET PARTITION (Cont.)

### /DIAGNOSTIC

Specifies a diagnostic partition. Diagnostic partitions are used for diagnostic functions on multiprocessor systems. The specified partition is created without checking the top of memory.

### /SYSTEM

Specifies a system partition. 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.

### 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. Because this is the default, the user need not have specified the /SYSTEM qualifier in the command.

```
$ SET NOPARTITION (RET)
Partition name? TOMPAR (RET)
$ SHOW PARTITION: TOMPAR (RET)
SET -- Partition not in system
```

This example eliminates the partition TOMPAR.

### Notes

Partitions are one of the key elements in tuning larger RSX-11M/M-PLUS systems and making the best use of available memory.

You can specify the partition you want 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 can display the name, size, address, and type of one or more partitions with the SHOW PARTITIONS command (Section 7.14.2). You can display the allocation of partitions in memory graphically with the SHOW MEMORY command (Section 8.3).

## SET PARTITION (Cont.)

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.

### Error Messages

#### SET---Alignment error

**Explanation:** The 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:** First, check command for proper values, and then reenter the command.

#### SET---Nonexistent memory

**Explanation:** Command attempted to define a partition in nonexistent memory.

**User Action:** First, check command for proper values, and then, reenter the command.

#### SET---Partition already exists

**Explanation:** Command attempted to define a partition with a name already in use.

**User Action:** Use another name.

#### 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.

# SHOW PARTITIONS

## 7.14.2 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 partype

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

The elements in both of these displays are as follows:

#### parname

Specifies the name of the partition about which you want information to be displayed. When you specify a partition, the display format is as follows:

PAR=parname:baseaddr:parsize:SYS

In this display, the address and size fields of the single partition display are 64-byte values (octal 100), while in the multiple partition display, byte values are displayed. See the examples.

#### 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.

#### partype

Specifies one of the following:

MAIN	Main partition
TASK	Dynamic task subpartition---not /TASK (user-controlled)
DRIVER	Device driver partition



## SHOW PARTITIONS (Cont.)

RO COM      Read-only common partition  
 RW COM      Read-write common partition  
 DEVICE      Partition that is a common mapped to the I/O page  
 SEC POOL    Partition that is a secondary pool

contentname

Indicates one of the following:

[taskname]      An inactive resident task  
 <TASKNAME>      An active task  
 (dd:)            A specific device driver  
 +xxxxx+          The name of the first installed task that is  
                   associated with an unnamed common, usually the  
                   read-only segment of a multiuser task  
 !yyyy!            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 1000(octal) bytes. See next example.

```
$ SHOW PARTITION: SYSPAR 
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 one hundred 64-byte blocks.

### Notes

SHOW PARTITIONS unadorned is the equivalent of the MCR PAR command.

SHOW PARTITIONS with a partition name is the equivalent of the MCR SET /PAR=parname command.

# SHOW TASKS

## 7.14.3 SHOW TASKS

You can display information about active and installed tasks on the system in a number of ways. Section 7.14.3.1 describes how static information about tasks active on your terminal can be displayed in full or brief format. Section 7.14.3.2 explains how static information about installed tasks can be displayed in full or brief format. Section 7.14.3.3 explains how static information about tasks installed from a specified device can be displayed. Section 7.14.3.5 explains the dynamic display of information about active tasks and Section 7.14.3.6 explains the dynamic display of the Active Task List. Section 7.14.3.7 explains the display of the LUNs of an installed task.

### 7.14.3.1 SHOW TASKS/ACTIVE

SHOW TASKS/ACTIVE displays information about active tasks in brief or full format.

#### Brief Format

```
SHOW TASKS/ACTIVE[:ttnn:][[/qualifier[s]]]
```

#### Command Qualifiers

```
/BRIEF
/ALL
```

#### Parameter

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 Qualifiers

```
/BRIEF
```

Specifies that you want to use the brief format to display information about active tasks. This is the default and need not be specified.

The brief format displays the task names and the originating terminal (in parentheses) for each task.

```
/ALL
```

Specifies that you want 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... (TT5:)
SHOT5 (TT5:)
```

This example shows the brief display from SHOW TASKS/ACTIVE. In this case, the tasks are MCR..., the central command dispatcher, and SHOT5, the SHOW task itself.

The terminal from which the task is running is also displayed.

```
$ SHOW TASKS/ACTIVE/ALL (RET)
LDR... (CO0:)
RMDEMO (TT0:)
SHOH2 (HT2:)
NETACP (CO0:)
MCR... (TT13:)
SHOT5 (TT5:)
DCLT5 (TT5:)
DB0FCP (CO0:)
.
.
.
EDTT2 (TT2:)
EDTT3 (TT3:)
```

This example shows the brief display from SHOW TASKS/ACTIVE/ALL.

The terminal from which the task is running is also displayed.

### Full Format

```
SHOW TASKS/ACTIVE/FULL [taskname]

SHOW TASKS[: taskname] /ACTIVE/FULL
```

### Parameter

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-buffiocount EFLG-eventflags PS-pswval
PC-pcval REGS 0-6 rlval r2val r3val r4val r5val r6val
```

## SHOW TASKS (Cont.)

The elements in this display are as follows:

taskname	The name of the task being shown.
tcbaddr	The physical address of the Task Control Block (TCB).
parname	The name of the task's partition.
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.
statusflags	The task's status flags. These are identified in Table 7-2.
ttnn	The initiating terminal.
iocount	The decimal I/O count for the task.
buffiocount	The decimal count of I/O buffered by the terminal driver and Executive.
eventflags	Local event flags.
pswval	The Processor Status Word.
pcval	The Program Counter.
rval	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.

# SHOW TASKS (Cont.)

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	TS.CHK	Not checkpointable
CIP	TS.CIP	Blocked for checkpoint in progress
CKD	TS.CKD	Checkpointing disabled
CKP	TS.CKP	Checkpointed
CKR	TS.CKR	Checkpoint request pending
CLI	T3.CLI	Command Line Interpreter
CMD	T3.CMD	Task is executing a CLI command
DSP	T4.DSP	Task was built for user-mode I- and D-space
DST	T2.DST	ASTs disabled
-EXE	TS.EXE	Not executing
FXD	T2.FXD	Fixed in memory
GFL	T3.GFL	Task has own group global event flags locked
HLD	TS.HLD	Half-loaded, awaiting pool for task header
HLT	T2.HLT	Being terminated
LDD	T4.LDD	Task's load device has been dismounted
MCR	T3.MCR	Activated by MCR
MSG	TS.MSG	Aborted, waiting for TKTN message
MUT	T4.MUT	Task is multiuser task
NET	T3.NET	Network protocol level
NSD	T3.NSD	Cannot receive data (no send data allowed)
OUT	TS.OUT	Out of memory
-PMD	T3.PMD	Suppress PMD on SST abort
PRO	T4.PRO	Task is (or should be) a prototype task
PRV	T3.PRIV	Privileged
PRV	T4.PRIV	Task was privileged but has cleared T3.PRIV
RDN	T3.RDN	I/O being run down
REM	T3.REM	Remove on exit
REX	T2.REX	Abort AST effected or in progress
ROV	T3.ROV	Resident overlays
RST	T3.RST	Restricted--used by layered software
SEF	T2.SEF	Stopped for event flag
SLV	T3.SLV	Slaved
SNC	T4.SNC	Task uses commons for synchronization
SPN	T2.SPN	Being suspended
SPNA	T2.SPN	Suspended prior to AST
STP	T2.STP	Stopped
STPA	T2.STP	Stopped prior to AST
SWS	T3.SWS	Reserved for software services
WFR	T2.WFR	In a wait-for state
WFRA	T2.WFR	In a wait-for state prior to AST
XHR	none	Task has an external header

# SHOW TASKS (Cont.)

## 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 Crash Dump Analyzer Reference Manual for more information. The CDA manual has a full explanation of the contents of the TCB.

## Examples

```
$ SHOW TAS/ACT/FULL TT7 (RET)
TV 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
BEGS 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
BEGS 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 - TTO: 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
```

# SHOW TASKS (Cont.)

```

DB00F1 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
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.

## 7.14.3.2 SHOW TASKS/INSTALLED -

SHOW TASKS/INSTALLED displays information about installed tasks in either brief or full format.



## SHOW TASKS (Cont.)

### Format

SHOW TASKS[:taskname]/INSTALLED[/qualifier]

### Command Qualifiers

/BRIEF  
/FULL

### Parameter

taskname

Specifies the task for which you want 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 ddnn:-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)
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 the task is to be loaded
lbn	The logical block number of the task's 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's irrelevant, 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.

## SHOW TASKS (Cont.)

/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 7.14.3.1 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
F11MSG 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. The task LDR... ( the loader ) is fixed in memory. The task EVP... is checkpointed. The final entry shown, MCR..., is the system command dispatcher. See next example.

```
$ 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 BEGS 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.

## SHOW TASKS (Cont.)

### 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 in real time as it is updated. See Section 7.14.3.5.

### 7.14.3.3 SHOW TASKS/INSTALLED/DEVICE -

The SHOW TASKS/INSTALLED/DEVICE command 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 LB00:-00502750
```

This display contains the following 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

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.

## SHOW TASKS (Cont.)

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.

### 7.14.3.4 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 hardcopy 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.

### 7.14.3.5 SHOW TASKS:taskname/DYNAMIC -

SHOW TASKS:taskname/DYNAMIC displays a moving picture of the task's activity on a video terminal or a snapshot display on a hardcopy terminal.

#### Format

SHOW TASKS:taskname/DYNAMIC[/qualifier]

#### Command Qualifiers

/RATE:n

#### Argument

taskname

Specifies the task you want to inspect.

#### Qualifiers

/DYNAMIC

Specifies that you want the moving display. This function is performed by the RMD task. RMD has the four display pages:

1. Task, which displays a task header
2. Active, which displays all or part of the Active Task List
3. Memory, which displays the contents of memory
4. Help, which provides help on RMD

## SHOW TASKS (Cont.)

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 8.1.9 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, where the named task will still be displayed.

/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 by pressing ESC and entering a new rate. The rate setting returns to 1 as soon as you change pages. If you want a slower rate, you must reset the rate each time you change the page.

Use this qualifier to slow down the display.

Task headers can change more often than once per second, but once per second is the most rapid rate that can be specified in a command.

### Example

\$ SHOW TASK:CORT5/DYNAMIC (RET)

The screen is cleared and the following display appears:

```

RSX-11M-PLUS V3.0 BL24 512K                      14-MAY-85
18:53:11

Task:  CORT5  Partition:  GEN STATUS:  -CHK  -PMD  REM  PRV  CLI
Own:   TT5:   I/O = 0.    DPri:  160 Pri:  160 Spri: 155 Len:
000356000

RO = 022120 R1 = 000764 R2 = 002230 R3 = 000764 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

Generally, this display is a useful tool for understanding how tasks run the system. However, you must have a thorough knowledge of how tasks run in the RSX-11M-PLUS 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 noting which files are open. You may also be interested in observing the Task Builder at work.

### 7.14.3.6 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
```

## SHOW TASKS (Cont.)

### 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.

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 so that it fits on one screen.

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 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 want 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.

### Examples

\$ SHOW TASKS/ACTIVE/DYNAMIC (RET)

## SHOW TASKS (Cont.)

The screen is cleared and the following display appears:

```

RSX-11M-PLUS V3.0 BL24      512K      14-MAY-85 19:08:55

  Name   Length   TI      Pri    I/O Status flags
LDR...  00002500  CD0      248.   0. -CHK FXD STP -PMD PRV NSD
RMDemo  00026700  TT0      225.   0. -CHK WFR -PMD PRV MCR
SHOHO   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
DBOFCP  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

```

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. Because 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-PLUS V3.0 BL24      512K      14-MAY-85 14:57:55

  Name   Length   TI      Pri    I/O Status flags
SHOTS   00026700  TT5      0.      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 8.1.9 for more information on RMD.

### 7.14.3.7 SHOW TASKS/LOGICAL UNITS -

SHOW TASK:taskname/LOGICAL UNITS displays the static logical unit number assignments (LUNs) for an installed task.

### Format

SHOW TASKS:taskname/LOGICAL UNITS



## SHOW TASKS (Cont.)

### 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.

### Examples

```
$ SHOW TASK: TEWKSY/LOGICAL UNITS (RET)
DB1:  1.
DB1:  2.
DB1:  3.
DB1:  4.
TI0:  5.
CL0:  6.
```

This example shows the display.

```
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. (See Chapter 5.)

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/DYNAMICcommand. (See Section 7.14.3.3.)

For more information on logical units, see Chapter 5. See also the RSX-11M/M-PLUS and Micro/RSX Task Builder Manual.

SHOW TASK/LOGICAL UNITS is the equivalent of the MCR LUN command.

Tasks can dynamically alter their LUN assignments through the ALUN\$ Executive directive.

## SHOW TASKS (Cont.)

### 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

### 7.14.4 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 want to assign to the task. Priority numbers are in the range of from 1 through 250. The argument 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 want 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)
TT7  066734  GEN 064244 01676500-02062100 PRI - 199. DPRI - 199.
$ SHOW TASK ACT FULL TV (RET)
      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
```

## SET PRIORITY (Cont.)

This example shows the effects of SET PRIORITY. The user first displayed the priority of task TT7 with the SHOW TASKS/FULL/ACTIVE command. The user then issued the SET PRIORITY command 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 processed. The user then attempted to change the task's priority to 255(decimal), 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/FULL/ACTIVE instead.

SET PRIORITY is the equivalent of the MCR ALT command.

## SHOW CLOCK\_QUEUE

### 7.14.5 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
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 2 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 the SHOW QUEUE command.

# SHOW COMMON

## 7.14.6 SHOW COMMON

SHOW COMMON displays the names of resident commons installed in the system, their PCB addresses, the number of attached tasks, and the status of each common. You can also display information about a single common, which optionally includes a list of tasks attached to the common.

### Format

SHOW COMMON[:commonname] [/TASK]

### Parameters

commonname

Specifies a common name about which information is to be displayed. If you do not name a common, information about all commons in the Common Block Directory is displayed in the following format:

commonname pcbaddr taskcount statusbits

The elements in this display are as follows:

commonname	The name of the common.
pcbaddr	The address of the Partition Control Block (PCB).
taskcount	The number of tasks mapped to the common.
statusbits	A list of common region status bits that are set. Usually, the status bits will indicate that the common region is either fixed in memory (FXD) or out of memory (OUT).

Here is a list of status bits and their meanings:

Status Bit	Meaning
CAF	Checkpoint allocation failure
-CHK	Not checkpointable
CKP	Checkpoint in progress
CKR	Checkpoint requested
COM	Library or common
DEL	Marked for delete on last detach
DRV	Driver common
FXD	Fixed in memory
LIO	Long I/O
LFR	Last load failed
NSF	Not shuffleable
OUT	Out of memory
PER	Parity error
PIC	Position independent
RON	Read-only common

## SHOW COMMON (Cont.)

### Command Qualifier

/TASK

Specifies that you want a list of tasks attached to a particular common region showing the number of times each task is mapped to the common (mapping count).

### Examples

```
$ SHOW COMMON (RET)
BASIC2 050644 0 STATUS: COM
DEPRES 056340 0 STATUS: COM
DSMBUF 053630 5 STATUS: COM PIC
DSMSUP 064164 2 STATUS: COM
FCSFSL 117210 17 STATUS: COM
TTCOM 034620 1 STATUS: FXD COM
```

This example shows the display from the unadorned SHOW COMMON command. Currently, six commons are installed in the system.

```
$ SHOW COMM: DSMBUF (RET)
DSMBUF 053630 5 STATUS: COM PIC
```

This example shows the display when a single common is named.

```
$ SHO COM:DSMBUF/TASK (RET)
DSMBUF 053630 5 STATUS: COM PIC
      DSMIO 3
      DSMGAR 1
      DSMT61 1
```

This example shows the display when a list of tasks is requested for a single common.

### Notes

SHOW COMMON is the equivalent of the MCR CBD command.





## 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 and library UFDs, 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.

## SET [DAY]TIME

### 8.1.1 SET [DAY]TIME

**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).

mm/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.

#### 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. The parameter ss defaults to 00.

#### Examples

```
$ SET TIME   
Time? 8:05 
```

This example sets the time at 8:05 on the current day.

```
$ SET DAY   
Date? 16-JUN-85 
```

This example sets the date at June 16, 1985.      The time setting is unchanged.

## SET [DAY]TIME (Cont.)

```
$ SET DAY   
Date? 06/14/85 17:14 
```

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-85 04:00:00   
$ SET TIME 04:00:00 06-JUN-85 
```

The examples are equivalent. Both set the date at June 6, 1985, 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

### 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-85
```

This example shows the date as June 6, 1985 and the time as 4:00:22.

## SET SYSTEM

## 8.1.3 SET SYSTEM

SET SYSTEM establishes certain characteristics of the system. It is a privileged command.

## Format

```
SET SYSTEM[/qualifier]
```

## Command Qualifiers

```
/EXTENSION LIMIT:n
/[NO]LOGINS
/NETWORK_UIC
/PACKETS:n
/POOL:top:max:total
/POOL/LIMITS:arg
                HIGH=n
                LOW=n
                MINIMUM_SIZE=n
                TASK_PRIORITY=n
/DIRECTORY:[directory]
```

## Command Qualifiers

```
/EXTENSION_LIMIT:n
```

Establishes the maximum size to which a task can extend itself by means of the EXTEND TASK directive. The argument n can be entered in all the formats used in the SET PARTITION command. See Section 7.14.1.

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 rebootstrap the system to log in again unless there is another privileged user logged in.

```
/NETWORK_UIC
```

Specifies the directory 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. The argument n can range from 0 through 15. The default is 5. The value is usually set as a part of the startup procedure.

This command is equivalent to the SET /MAXPKT command in MCR and VMR.

## SET SYSTEM (Cont.)

SHOW SYSTEM/PACKETS displays this limit and also the number of packets currently available. See Section 8.1.4.

/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

Set 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.
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.

/DIRECTORY:[directory]

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 qualifier for SET SYSTEM.

See the discussion in Section 8.2 for information on the use of this command on RSX-11M-PLUS systems.

## SET SYSTEM (Cont.)

## 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  
 /SECONDARY\_POOL

#### 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. SHOW LIBRARY/DIRECTORY displays the current library directory. See Section 8.2 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).



**SHOW SYSTEM (Cont.)****/NETWORK\_UIC**

Displays the current network directory.

**/PACKETS**

Displays the maximum number and the current number of available I/O packets in the following format:

MAXPKT=m.:c.

where m is the maximum number of 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: The first location available for user partitions

lfb: The largest free block in decimal words

tps The total pool space in decimal words

The SHOW MEMORY command (Section 8.3) 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

## SHOW SYSTEM (Cont.)

### /SECONDARY\_POOL

Displays information about secondary pool. The display is in the following format:

SECPOL=free:size:pct

The elements are as follows:

free    The amount of free secondary pool in 32-word blocks

size    The size of secondary pool in 32-word blocks

pct    The percentage of free blocks in secondary pool.

### 8.2 SYSTEM AND LIBRARY DIRECTORIES

On RSX-11M-PLUS systems, the system image file and all privileged system tasks are kept in the system directory on pseudo device LB:. You can display the directory with the SHOW SYSTEM/DIRECTORY command. Privileged users can change the directory with the SET SYSTEM/DIRECTORY command. All nonprivileged system tasks and utilities are kept in the library directory on pseudo device LB:. You can display the current setting for this directory with the SHOW LIBRARY/DIRECTORY command. Privileged users can change the directory with the SET LIBRARY/DIRECTORY command.

When you issue a RUN or INSTALL command and precede the file specification (the task image file name) with a dollar sign (\$), you are instructing the system that the task you want to run or install is in one of these directories. The system directory is searched first and then the library directory.

## SET LIBRARY/DIRECTORY

### 8.2.1 SET LIBRARY/DIRECTORY

The SET LIBRARY/DIRECTORY command establishes the identity of the system library directory on an RSX-11M-PLUS system. The command does not create a directory. The directory must exist before you issue the command.

See the discussion in Section 8.2 for the relationship between SET LIBRARY/DIRECTORY and SET SYSTEM/DIRECTORY.

#### Format

SET LIBRARY/DIRECTORY:[directory]

## SHOW LIBRARY


### 8.2.2 SHOW LIBRARY

The SHOW LIBRARY command displays the current library directory on RSX-11M-PLUS systems. The display specifies LIBUIC, but the library directory rarely has a UIC associated with it; it is a directory.

#### Format

```
SHOW LIBRARY
```

#### Example

```
$ SHOW LIBRARY   
LIBUIC= [3,54]
```

This example shows the system library directory as [3,54].

## SHOW USERS

## 8.3 SHOW USERS

SHOW USERS displays the terminal number, login UIC, and default directory for all logged-in users. If Resource Accounting is supported and is running, SHOW USERS also shows the date and time of login, the number of tasks active on the terminal, and the user name of the person logged in.

## Format

SHOW USERS

## Examples

\$ SHOW USERS (RET)

TT2:	[7,40]	[7,40]
TT5:	[301,370]	[301,370]
TT7:	[200,1]	[200,1]
TT16:	[7,42]	[7,42]
TT20:	[200,20]	[7,37]
TT51:	[1,2]	[7,26]
TT54:	[1,10]	[303,5]
HT2:	[7,43]	[7,43]

This example shows the SHOW USERS display for an RSX-11M-PLUS system without Resource Accounting active. The first column identifies the terminal. The second column is the default directory. Column three is the login UIC. HT2: is a DECnet host terminal.

\$ SHOW USERS (RET)

TT7:	[7,40]	[7,40]	7-MAY-8510:57	0	J. ANGEL
TT10:	[303,5]	[7,100]	7-MAY-8511:10	2	W.WRITERS
TT14:	[7,25]	[7,25]	7-MAY-8513:56	0	D. DUCK
TT15:	[1,1]	[7,50]	7-MAY-85 11:06	0	G. VOSKA
TT16:	(314,362]	[7,14]			
TT17:	[UYAK]	[7,60]	7-MAY-85 11:15	0	P. STROM
TT20:	[12,10]	[7,24]	7-MAY-85 10:47	0	D.UYAK
TT21:	[1,6]	[7,13]	7-MAY-85 10:50	1	P. KROBAR
TT27:	[303,2]	[303,2]	7-MAY-85 12:55	0	C.FUTILE
TT30:	[FRINI]	[303,10]	7-MAY-85 12:33	0	J. FRINI
TT32:	[VEISS]	[7,11]	7-MAY-85 14:14	1	P. VEISS
TT33:	[7;16]	[7,16]	7-MAY-85 10:56	0	S. THOMS
TT34:	[1,24]	[7,21]	7-MAY-85 11:08	0	N. EPPS
TT35:	[KNOCKS]	[301,370]	7-MAY-85 12:36	1	D. KNOCKS
TT36:	[7,302]	[7,302]	7-MAY-85 10:55	0	S. ADAMS
TT37:	[GAZLE]	[303,14]	7-MAY-85 13:01	1	J. GAZLE
TT41:	[303,12]	[303,12]	7-MAY-85 12:36	0	J. ALNESE
TT42:	[304,325]	[7,325]	7-MAY-85 10:56	0	P. BEZ
TT43:	[303,23]	[303,23]	7-MAY-85 11:17	0	L. KROGAN
TT45:	[7,43]	[7,43]	7-MAY-85 14:35	0	J. MAN
VT3:	[ANGLE]	[303,5]	7-MAY-85 15:04	2	M. ANGLE

## SHOW USERS (Cont.)

This example gives the SHOW USERS display for an RSX-11M-PLUS system with Resource Accounting. The elements of the display are as follows:

Column	Function
1	Identifies the user's TI:
2	Identifies the current default directory
3	Identifies the login UIC
4	Gives the date and time of login
5	Identifies the number of tasks currently active
6	Identifies the user name

If Resource Accounting is not active or running at the time of login, the last three columns are not displayed. In this example, the user on TT16: logged in when Resource Accounting was not active.

The SHOW MEMORY command 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).

### Format

### Example

\$ SHOW MEMORY RET

8-15

## SHOW MEMORY (Cont.)

- 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 Time elapsed in units of days, hours, and minutes since the system was last bootstrapped.
- ⑤ Current date and time.
- ⑥ Name of the task that is currently executing or, if none is executing, \*IDLE\* (Executive executing the idle loop).
- ⑦ 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."
- ⑧ Pool (dynamic storage region) information in the following 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 fragments in the pool free list

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 booted.

- ⑨ Secondary pool information in the following format:

SECPool=A:B:C%

A Number of free blocks in secondary pool

B Total number of blocks in secondary pool

C% Percentage of secondary pool that is free

The second line records the worst case of secondary pool since you invoked RMD. This line is most useful if RMD has been running on a slaved terminal since the system was last booted.



**SHOW MEMORY (Cont.)**

- ⑩ Partitions in the system using the following format:

partitionname:type

where type is one of the following:

D System-controlled (dynamic) partition

P Secondary pool partition

- ⑪ 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.
- ⑫ 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.

- ⑬ 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.

## SHOW MEMORY (Cont.)

- 14 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 hangs) 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.

Other pages in the RMD display are invoked by the SHOW TASKS/DYNAMIC command (Section 7.14.3.4). 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-PLUS 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 preceded by 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.

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-PLUS 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	FORTTRAN	S	SHOW

#### ABORT -- Chapter 7

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:. Privileged users can abort any task by using the /TERMINAL qualifier.

## QUICK REFERENCE

### Formats

ABORT[/COMMAND] [/qualifier[s]] commandname

ABORT/TASK[/qualifier] taskname

### Command Qualifiers

/COMMAND	!Default
/[NO]POSTMORTEM	
/TASK	
/TERMINAL:ttnn:	!Privileged

### ALLOCATE -- Chapter 4

ALLOCATE declares a specified device to be a private device. The format for the logical device name is the same as for other device names: two characters, an octal unit number, and a colon. If you omit the unit number and colon, the first available device of that class is allocated.

### Format

ALLOCATE[/qualifier[s]] dd[nn:] [logicalname]

### Command Qualifiers

/TERMINAL:ttnn:	!Privileged
/TYPE:devicetype	!RM05, RK06, RL02, and so on

### ANALYZE/CRASH\_DUMP -- RSX-11M/M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual

The 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[/qualifer[s]]

### 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	

## QUICK REFERENCE

### Filespec 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
/KERNEL: (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])
    DATA: (NAME:name[,START:n,END:n])
    INSTRUCTION: (NAME:name[,START:n,END:n])
```

### ANALYZE/MEDIA -- Chapter 5

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

### Format

ANALYZE/MEDIA[/qualifier[s]] ddnn:

### Command Qualifiers

```
/ALLOCATE:label                                !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
```

## QUICK REFERENCE

### APPEND -- Chapter 3

APPEND appends to an existing sequential file records from one or more sequential files.

#### Format

APPEND[/qualifier[s]] infile[,s] outfile

#### Command Qualifiers

/EXCLUDE:filespec	!Filespec can include wildcards.
/NOWARNINGS	!Suppresses error messages
/REWIND	!Tape only. Rewinds tape before beginning
/SHARED	!Permits others to access file while you append 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-mmm-yy	!From given day through given day
/TODAY	!Today only

### ASSIGN -- Chapter 5

ASSIGN equates a logical name to a physical device name, to all or part of a file specification, or to another logical name. ASSIGN checks the syntax of an equivalence name that is either a device or file specification. All references to the logical name are resolved by the operating system.

#### Format

ASSIGN[/qualifier[s]] equivalence\_name logical\_name

#### Command Qualifiers

/FINAL	!Privileged
/GROUP[:g]	!UIC group number
/LOCAL	!Default
/LOGIN	!Privileged
/GLOBAL	!Privileged
/SYSTEM	!Privileged; synonym for /GLOBAL
/TERMINAL:ttnn:	!Privileged
/TRANSLATION:ATTRIBUTES=TERMIHNAL	!Privileged; synonym for /FINAL

### ASSIGN/QUEUE -- 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 queue\_name processor\_name

## QUICK REFERENCE

### ASSIGN/REDIRECT -- Chapter 5

ASSIGN/REDIRECT redirects output from one physical device to another. You can also redirect output from a physical device to a pseudo device or the reverse. Privileged.

#### Format

ASSIGN/REDIRECT oldddnn: newddnn:

### ASSIGN/TASK -- Chapter 7

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 -- Chapter 5

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.

#### Format

BACKUP[/qualifier[s]] sourceddnn:[filespec[s]] destinationddnn:

#### Command Qualifiers

##### Group 1: Selective Backup and Restore

/CREATED:arg  
    AFTER:(dd-mmm-yy hh:mm)  
    BEFORE:(dd-mmm-yy hh:mm)  
/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

## QUICK REFERENCE

### Group 2: Initialization

/ACCESSED:n	!n is default number of FCBs per !volume.
/BAD_BLOCKS: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	
800	
1600	!Default density is 800 bpi.
/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

### 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.



## QUICK REFERENCE

### BROADCAST -- Chapter 3

BROADCAST displays a specified message at one or more terminals.

#### Formats

BROADCAST ttnn: message

BROADCAST username message

BROADCAST @indirectspec

BROADCAST[/qualifier] message

#### Command Qualifiers

/ALL	!Privileged
/LOGGED_IN	!Privileged

### CANCEL -- Chapter 7

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 -- Chapter 7

CONTINUE resumes execution of a previously suspended task. The task name defaults to terminal TTnn:.

#### Format

CONTINUE[/qualifier] [taskname]

#### Command Qualifier

/TERMINAL:ttnn:	!Privileged
-----------------	-------------

### CONVERT -- Chapter 4

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.

#### Format

CONVERT[/qualifier[s]] infile outfile

## QUICK REFERENCE

### Command Qualifiers

/[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-11 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 -- Chapter 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. If you intend to do multiple file copies, see the COPY command description.

### Format

COPY infile[s] outfile

### Command Qualifiers

/ALLOCATION:n	!Specifies n blocks of contiguous !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.
/EXCLUDE:filespec	!Filespec can include wildcards.
/NONEW_VERSION	!Suppresses automatic increment !of version numbers.
/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	!No magtape.
/REWIND	!Magtape only. Rewinds tape before !starting operation.
/SHARED	!Permits others to access file !while you copy 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-mmm-yy	!From given day through given day.
/TODAY	!Today only.

## QUICK REFERENCE

### CREATE -- Chapter 4

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, press **CTRL/Z**. See also CREATE/DIRECTORY.

#### Format

CREATE filespec

### CREATE/DIRECTORY -- Chapter 4

CREATE/DIRECTORY creates a directory 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:][directory]

#### 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 -- Chapter 5

DEALLOCATE counteracts ALLOCATE. It frees a private device for access by others. To deallocate a device, the device must not be mounted.

#### Format

DEALLOCATE[/qualifier] ddnn:

#### Command Qualifiers

/ALL	!Frees all devices allocated by
	!TI;; do not specify ddnn:.
/DEVICE	!Non-operational
/TERMINAL:ttnn:	!Privileged

### DEASSIGN -- Chapter 5

DEASSIGN deletes logical name assignments. DEASSIGN counteracts both the ASSIGN and DEFINE commands.

#### Format

DEASSIGN[/qualifier[s]] logical\_name

## QUICK REFERENCE

### Command Qualifiers

/ALL	!Combine with any other qualifier
/GROUP[:g]	!UIC group number
/LOCAL	!Default
/LOGIN	!Privileged
/GLOBAL	!Privileged
/SYSTEM	!Synonym for /GLOBAL
/TERMINAL:ttnn:	!Default is TI;; otherwise
	!privileged

### DEASSIGN/QUEUE -- 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 queue\_name processor\_name

### DEBUG -- Chapter 7

Debug forces a task on RSX-11M-PLUS to trap a debugger by setting the T-bit in the task's Processor Status Word. The task must have been built using the /DEBUG qualifier to the LINK command or have issued an Executive directive specifying a debugger. See main text.

Nonprivileged users can issue this command for any nonprivileged task running from their own terminals. Privileged users can name any task, but the command must be issued from the same terminal the task was run from. The default task name is TTnn:.

#### Format

DEBUG [taskname]

### DEFINE -- Chapter 5

DEFINE equates a logical name to a physical device name, to all or part of a file specification, or to another logical name. All references to the logical name are resolved by the operating system.

Unlike ASSIGN, DEFINE does not check the syntax of an equivalence name that is either a device or file specification.

Please note that the DEFINE command primarily benefits the applications programmer.

#### Format

DEFINE[/qualifier[s]] equivalence\_name logical\_name

## QUICK REFERENCE

### Command Qualifiers

/FINAL	!Privileged
/GROUP[:g]	
/GLOBAL	!Privileged
/LOCAL	!Default
/LOGIN	!Privileged
/SYSTEM	!Privileged; synonym for /GLOBAL
/TERMINAL:ttnn:	!Privileged
/TRANSLATION_ATTRIBUTES=TERMINAL	!Privileged; synonym for /FINAL

### DELETE -- Chapter 4

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]]

### Command Qualifiers

/[NO]CONFIRM	!Synonym for /QUERY
/EXCLUDE:filespec	!Filespec can include wildcards.
/[NO]LOG	!Lists deleted files on TI:
/NOWARNINGS	!Suppresses error messages
/[NO]QUERY	!Queries before deleting;
	!/NO is default.
/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

### DELETE/DIRECTORY -- Chapter 4

DELETE/DIRECTORY deletes a directory on a Files-11 volume and removes its name from the volume's Master File Directory (MFD). Nonprivileged users can only delete directories on mounted volumes on their own private (allocated) device.

#### Format

DELETE/DIRECTORY [ddnn:][directory]

### DELETE/ENTRY -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

DELETE/ENTRY deletes QMG jobs by entry number (n).

#### Format

DELETE/ENTRY:n[/qualifier]

### Command Qualifiers

/FILE\_POSITION:n

## QUICK REFERENCE

### DELETE/JOB -- Chapter 4

See also 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 [[uic]] job name

#### Command Qualifier

/FILE\_POSITION:n

### DELETE/processor type -- RSX-11M/M-PLUS System Management Guide

DELETE/processor type 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.

#### Format

DELETE/processor type processor name

#### Processortypes

APPLICATIONS PROCESSOR

BATCH PROCESSOR

CARD READER

DEVICE

INPUT

PRINTER

PROCESSOR

!INPUT is synonym.

!PRINTER is synonym.

!CARD READER is synonym.

!DEVICE is synonym.

### DELETE/QUEUE -- RSX-11M/M-PLUS System Management Guide

Privileged. DELETE/QUEUE deletes queues in the Queue Manager subsystem by name. When you specify DELETE/QUEUE, you must also specify the /ERASE qualifier. See DELETE/JOB and DELETE/ENTRY to delete jobs from queues.

#### Format

DELETE/QUEUE queue name/ERASE

### DIFFERENCES -- Chapter 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.

DIFFERENCES infile1 infile2

## QUICK REFERENCE

### Command qualifiers

/CHANGE_BAR[:n]	!n is octal ASCII code of alternate !bar. Default n=041 (!).
/IGNORE:(arg[,s]) BLANK_LINES COMMENTS FORM FEEDS SPACING TRAILING_BLANKS	!Comments begin with ";". !Group of tabs and blanks = 1 !blank.
/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.

### DIRECTORY -- Chapter 4

DIRECTORY displays information on files in directories.

### 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 Qualifiers

/OUTPUT[:filespec]	!Names output file. TI: is default.
/PRINTER	!Output to LP0:

#### Other Qualifiers

	!Not used with /FREE or !/ATTRIBUTES
/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	!Suppresses error messages
/REWIND	!Magtape only; rewinds tape before !executing command

## QUICK REFERENCE

### DISMOUNT -- Chapter 5

DISMOUNT marks the volume mounted on the specified device as logically off line and disconnected from the file system. If you specify the volume label, the system verifies the label name to ensure that you are dismounting the correct disk.

#### Format

DISMOUNT ddnn: [label]

#### 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:ttnn:	!Privileged; dismounts volumes !mounted from another terminal.
/[NO]UNLOAD	!Affects DB:, DM:, and DU: devices; !see main text.

### EDIT[/EDT] -- Chapter 4

See also EDT Editor Manual.

EDIT or EDIT/EDT invokes EDT, the standard DIGITAL 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 -- Chapter 4

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



## QUICK REFERENCE

### Command Qualifiers

/[NO]AUDIT[: (arg[s])]	
POSITION:n	!n must be <=132; Default = 80.
SIZE:n	!n must be <14; default = 8.
/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 EDT/editor -- Chapter 4

EDIT/EDI invokes the Line Text Editor (EDI). EDT/editor invokes a user-specified editor (see the following list).

#### Format

EDIT[/qualifier] [edit-input]

### Command Qualifiers

/EDI	!Line Text Editor
/KED	!Unbundled KED editor
/K52	!VT52 version of unbundled
	!KED editor
/MAKE	!Unsupported TECO function
/MUNG	!Unsupported TECO function
/OUTPUT:filespec	!Use with /KED and /K52
/CREATE	!Use with KED and K52
/SOS	!Unsupported Son of Stopgap
/TECO	!Unsupported Text Editor and
	!Corrector
/USING:yyy	!Unsupported user editor

### FIX -- Chapter 7

FIX causes an installed task or region to be loaded and locked into memory. The taskname parameter can be the name of either the task or the region. Privileged.

#### Format

FIX[/qualifier] taskname

### Command Qualifiers

/READONLY_SEGMENT	!Read-only segment of multiuser
	!task
/REGION	!Common region

## QUICK REFERENCE

### HELP -- Chapter 3

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]] [parameter 1 ...parameter9]

#### Command Qualifiers

/CLI:cliname	!For alternate CLIs.
/DCL	!DCL help; default for DCL
	!terminals.
/FILE:filespec	!Names file containing help text.
/filename	!LB:[1,2] filename.HLP
/GROUP	!Help file is in [g,l]; g is your
	!group.
/MCR	!MCR help; default for MCR
	!terminals.
/OUTPUT:filespec	!Help text to file; default
	!/OUTPUT:TI:.

### HOLD/ENTRY -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

HOLD/ENTRY holds a QMG job in its queue by entry number (n).

#### Format

HOLD/ENTRY:n

### HOLD/JOB -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

HOLD/JOB holds a QMG job in its queue by queue name and job name.

#### Format

HOLD/JOB queue name [[uic]]job name

### INITIALIZE -- Chapter 5

INITIALIZE produces a volume in Files-11 format. See also INITIALIZE/UPDATE. You must mount the volume /FOREIGN. Nonprivileged users must allocate the device before it can be initialized. See also other forms of the INITIALIZE command described in the following entries.

#### Format

INITIALIZE[/qualifier[s]] ddnn: volumelabel

## QUICK REFERENCE

### Command Qualifiers

/ACCESSED:n	!Number of directories accessed !simultaneously.
/BAD_BLOCKS:arg	
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.
/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"	!Magtape only: limits access. !c can be A--Z, 0--9, !and ! " % ' ( ) + , - . / : != & > < _ ? or ; !Default is blank.
/MAXIMUM_FILES:n	
/OWNER:[uic]	
/PROFESSIONAL	!Initialize disk for Professional !300 series.
/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 type (Input) -- RSX-11M/M-PLUS System Management Guide

This form of the INITIALIZE command creates, names, and starts an input spooler or card-reader processor. Privileged. See next entry to initialize an output despooler.

#### Format

INITIALIZE/processor type processorname[/qualifier[s]]

#### Processor Types

CARD_READER	!INPUT is synonym.
INPUT	!CARD_READER is synonym.

## QUICK REFERENCE

### Command Qualifiers

/BATCH\_QUEUE:queuname  
/CONSOLE:ddnn:  
/PRINTER\_QUEUE:queuname

### INITIALIZE/processor type (Output) -- RSX-11M/M-PLUS System Management Guide

This form of the INITIALIZE command creates, names, and starts an output despooler or batch processor. Privileged. See previous entry to initialize an input spooler.

### Format

INITIALIZE/processor type processorname[/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/QUEUE -- RSX-11M/M-PLUS System Management Guide

INITIALIZE/QUEUE creates, names, and starts a queue in the Queue Manager subsystem.

### Format

INITIALIZE/QUEUE queuname[/qualifier]

### Command Qualifiers

/BATCH	
/PRINTER	!Default
/NOWARNINGS	

### INITIALIZE/UPDATE -- Chapter 5

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 must mount the volume /FOREIGN.

### Format

INITIALIZE/UPDATE[/qualifier[s]] ddnn: volumelabel

## QUICK REFERENCE

### Command Qualifiers

/ACCESSED:n	!Extend full files by n blocks.
/DENSITY:arg	!Default protection for files on
HIGH	!volume.
LOW	!Changes volume label.
/EXTENSION:n	!Maximum number of files on volume.
/FILE_PROTECTION:code	!Initialize disk for Professional
/LABEL:newvolumelabel	!300 series.
/MAXIMUM_FILES:n	!Protection for volume.
/OWNER:[uic]	
/PROFESSIONAL	
/PROTECTION:code	
/[NO]SHOW	
/WINDOWS:n	!Mapping pointers to file windows;
	!default n=7.

### INSTALL -- Chapter 7

INSTALL includes a task in the System Task Directory, thus making it known to the system. Privileged.

#### Format

INSTALL[/qualifer[s]] [\$]filespec

### Command Qualifiers

/[NO]CHECKPOINT	!n (decimal) additional words of
/COMMAND:"taskcommand"	!address space.
/EXTENSION:n	!Installing a CLI? Default is NO.
/[NO]INTERPRETER	!Install read-only portion.
/MULTIUSER_PARTITION:parname	
/PARTITION:parname	
/[NO]POSTMORTEM	
/PRIORITY:n	!0--250.
/[NO]READONLY_COMMON	!Install common as read-only.
	!Default is NO.
/[NO]RESIDENT_HEADER	!/NORESIDENT_HEADER is
	!external header;/RESIDENT is
	!no external header.
/[NO]SLAVE	!/NOSLAVE is default.
/TASK_NAME:taskname	!1--6 characters.
/TRANSLATION_ROUTINE:n	
/UIC:[uic]	
/[NO]WRITEBACK	

### LIBRARY -- Chapter 6

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 qualifers.

#### Format

LIBRARY[/operation] [/qualifer[s]]

LIBRARY @filespec

## QUICK REFERENCE

### LIBRARY/COMPRESS -- Chapter 6

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.

#### Formats

LIBRARY/COMPRESS[: (arg[,s])] libspec [newlibspec]

#### Arguments

GLOBAL:n	!Entry-point table entries.
MODULES:n	!Module-name table entries.
BLOCKS:n	!Size in 256-word blocks.

### LIBRARY/CREATE -- Chapter 6

LIBRARY/CREATE creates a library and optionally inserts one or more modules into it.

#### Formats

LIBRARY/CREATE[: (arg[,s])] [/qualifier[s]] libspec [infilespec[s]]

#### Arguments

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 -- Chapter 6

LIBRARY/DELETE deletes object modules from a library. You can delete as many as 15 modules with a single command. See LIBRARY/REMOVE for removing global symbols (entry points) from a library.

#### Format

LIBRARY/DELETE libspec module[,module[,s]]

### LIBRARY/EXTRACT -- Chapter 6

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.

## QUICK REFERENCE

### Format

LIBRARY/EXTRACT[/qualifier] libspec module[,s]

### Command Qualifier

/OUTPUT[:filespec]                   !Default output file is TI;; name a  
                                      !file when extracting object  
                                      !modules.

### LIBRARY/INSERT -- Chapter 6

LIBRARY/INSERT inserts modules from one or more files into a library.  
You can insert any number of files with a single command.

### Format

LIBRARY/INSERT libspec filespec[s]

### Command Qualifiers

/[NO]GLOBALS                         !Include globals in entry-point  
                                      !table.  
/SELECTIVE\_SEARCH  
/SQUEEZE

### LIBRARY/LIST -- Chapter 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 -- Chapter 6

LIBRARY/REMOVE removes global symbols (entry points) from a library.  
You can remove as many as 15 global symbols with a single command.  
See LIBRARY/DELETE for deleting object modules from a library.

### Format

LIBRARY/REMOVE libspec global[,global[,s]]

### LIBRARY/REPLACE -- Chapter 6

LIBRARY/REPLACE replaces a module in a library with a new module of  
the same name and deletes the old module.

## QUICK REFERENCE

### Format

LIBRARY/REPLACelibspec filespec[s]

### Command Qualifiers

/[NO]GLOBALS !Include globals in entry-point  
!table.  
/SELECTIVE\_SEARCH  
/SQUEEZE

### LINK -- Chapter 6

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

### Command Qualifiers

/ANCILLARY\_PROCESSOR[:n] !Task is ACP; n is 0,4, or 5.  
/[NO]CHECKPOINT:arg  
    SYSTEM !Checkpoints to [0,0]CORIMG.SYS.  
    TASK !Checkpoints to task image file.  
/CODE:(arg[,s])  
    CLI !CLI task.  
    DATA\_SPACE  
    EAE !Extended arithmetic element.  
    FAST\_MAP !Fast Mapping.  
    [NO]FPP !Floating-point processor. Default  
            is FPP.  
    PIC !Position-independent code.  
    POSITION\_INDEPENDENT !Synonym for PIC.  
/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.  
/[NO]EXTERNAL  
/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?  
/[NO]PRIVILEGED[:n] !Default is /NOPRIVILEGED.  
/[NO]RECEIVE  
/[NO]RESIDENT\_OVERLAYS  
/SAVE !Saves indirect file.  
/[NO]SEGREGATE !/NOSEG is default.



## QUICK REFERENCE

/SEQUENTIAL	
/SHAREABLE[:arg]	!Multiuser.
COMMON	
LIBRARY	
TASK	!Default.
/SLAVE	
/SLOW	
/SYMBOL TABLE[:filespec]	!Output .STB file.
/[NO]SYSTEM_LIBRARY_DISPLAY	
/[NO]TASK[:filespec]	!Default is /NOSYS. !Names task image file; !/EXECUTABLE !is synonym; names 1--6 characters; !/NOTASK means no task built. !Default.
/TKB	
/TRACE	
/[NO]WARNINGS	!/NO suppresses diagnostic !messages; /WARNINGS is default.
/[NO]WIDE	!Wide map.

### File Qualifiers

/[NO]CONCATENATE	
/DEFAULT_LIBRARY	!File to replace [001001]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 -- Chapter 3

LOGIN (or HELLO) grants access to a multiuser protection system and establishes your privileges as a system user.

### Format

LOGIN userid/password

### LOGOUT -- Chapter 3

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]

### Command Qualifier

/[NO]HOLD	!Holds remote line after logout; !/NO is default.
-----------	--

## QUICK REFERENCE

### MCR -- Chapter 1

MCR enters an MCR command from a DCL terminal without leaving DCL.

#### Format

MCR mcrcommand

### MOUNT -- Chapter 5

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

/CACHE:(option[,s])	!Controls data caching; see !RSX-11M/M-PLUS <u>System</u> !Management <u>Guide</u>
PAR=[main_pname:]subpname[:size]	
[NO]DIRECTORY	
[NO]OVERLAY	
[NO]VIRTUAL	
[NO]LOGICAL	
[NO]READ_AHEAD	
/NOCACHE	
/DEFAULT:arg	!Sets defaults for dismount; see !main text.
SAVE	
NOUNLOAD	
UNLOAD	
/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	!Displays volume information on !TI:.
/SYSTEM	!Synonym for /PUBLIC.
/[NO]WAIT	!Default is /NOWAIT.
/[NO]WRITE	

## QUICK REFERENCE

### Command Qualifiers for Files--11 Devices

/ACCESSED:n	!n is number of File Control !Blocks.
/EXTENSION:n	!Extends full files by n blocks.
/OWNER:[uic]	!Coordinates with file and volume !protection.
/UNLOCK	!Main use is with VFY.
/WINDOW:n	

### Command Qualifiers for ANSI and Unlabeled Tapes

See main text for these qualifiers.

/BLOCK_SIZE:n	
/CARRIAGE_CONTROL:arg	FORTTRAN LIST NONE
/DENSITY:arg	800 1600
/[NO]HDR3	
/[NO]LABEL	
/OVERRIDE:(arg,[,s])	ACCESSIBILITY EXPIRATION DATE IDENTIFICATION SET IDENTIFICATION
/RECORD_SIZE:n	
/TRANSLATE:arg	EBCDIC NONE UT1 UT2 UT3
/VOLUME_IDENTIFICATION:(volume-ID[,volume-ID[,s]])	

### PRINT -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

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.

## QUICK REFERENCE

/LENGTH:n	!Page length.
/[NO]LOWERCASE	
/NAME:jobname	!1--9 characters.
/[NO]TRANSFER	!Override on filespec.
/PAGE COUNT:n	!Limits pages in job.
/PRIORITY:n	!n is 1--150 nonprivileged; through !250 privileged; default n=50
/QUEUE:queue name	
/[NO]RESTART	
/[NO]UPPERCASE	

### File Qualifiers

/COPIES:n  
/[NO]DELETE  
/[NO]TRANSFER

### PURGE -- Chapter 4

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

/EXCLUDE:filespec	!Filespec can include wildcards.
/KEEP:n	
/[NO]LOG	!Lists files on TI: as deleted.
/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.

### RELEASE/ENTRY -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

RELEASE/ENTRY releases by entry number (n) a print or batch job that has been held in its queue.

### Format

RELEASE/ENTRY:n

## QUICK REFERENCE

### RELEASE/JOB -- Chapter 4

See also 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 [[uic]] job name

### REMOVE -- Chapter 7

REMOVE counteracts INSTALL. REMOVE takes a task name out of the System Task Directory. Privileged.

#### Format

REMOVE[/qualifier] task name

#### Command Qualifier

/REGION	!Takes name of region out of Common !Block Directory and partition !list.
/TRANSLATION_ROUTINE:n	!Removes an ACD

### RENAME -- Chapter 4

RENAME changes the name, type, or version number of an existing file. Note that you can specify wildcards in any the file specification fields, but you should be careful when using them so that you do not specify the wrong file. You should also be careful with version numbers.

#### Format

RENAME[/qualifier[s]] oldfilespec newfilespec

#### Command Qualifiers

/EXCLUDE:filespec	!Filespec can include wildcards.
/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

### REQUEST -- Chapter 3

REQUEST sends a message to the operator's console (CO:). You can optionally place quotation marks (") around the message.

#### Format

REQUEST message

## QUICK REFERENCE

### RUN Installed Task -- Chapter 7

RUN 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

/DELAY:nu	!n is the number of units and u is !the time unit: T,ticks; S,seconds; !M,minutes; H,hours. Privileged.
/INTERVAL:nu	!Privileged.
/SCHEDULE:hh:mm:ss	!Privileged.
/STATUS:arg	
COMMAND	!Return status from RUN command; !default. Nonprivileged.
TASK	!Return status from task being run; !see main text on both these !arguments. Nonprivileged.
/SYNCHRONIZE:u	!Synchronize on next T,S,M, or H. !Privileged.
/UIC:[uic]	!Privileged.

### RUN Uninstalled Task -- Chapter 7

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]] [\$]filespec

#### Command Qualifiers

/[NO]CHECKPOINT	!See main text.
/COMMAND:"taskcommand"	!n (decimal) additional words of !address space.
/EXTENSION:n	!Privileged
/PARTITION:parname	!Privileged
/[NO]POSTMORTEM	
/PRIORITY:n	!Privileged
/STATUS:arg	
TASK	!Return status from RUN command; !the default
COMMAND	!Return status from task being run; !see main text on both these !arguments. /STATUS can be !used separately or with /UIC
/TASK_NAME:taskname	!1--6 characters
/TIME_LIMIT:n[u]	!Arg is M (minutes) by default; can !also be S (seconds); 3M is !default.
/UIC:[uic]	!Privileged

## QUICK REFERENCE

### SET [DAY]TIME -- Chapter 8

SET [DAY]TIME sets the system date and time. You must specify at least one of the fields, in any order. Privileged.

#### Format

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

### SET DEBUG -- Chapter 1

SET DEBUG displays the MCR translation of any DCL command.

#### Format

SET DEBUG[/qualifier[s]]

#### Command Qualifiers

/[NO]EXECUTE	!Translate, then execute the
	!command. Default is /NOEXECUTE
/FULL	!Display logical symbols
	!and translation.

### SET DEFAULT -- Chapter 4

SET DEFAULT establishes your default device or directory, or both. If you are a privileged user and your terminal's default is /NONAMED\_DIRECTORY, then SET DEFAULT also establishes your User Identification Code.

#### Format

SET DEFAULT[/qualifier] [ddnn:][[directory]]

#### Command Qualifier

/[NO]NAMED_DIRECTORY	!See main text.
----------------------	-----------------

### SET DEVICE -- Chapter 5

SET DEVICE establishes certain device attributes. Privileged.

#### Format

SET DEVICE:ddnn:/qualifier[s]

#### Command Qualifiers

/CACHE:(option[,s])	!Modifies data caching; see
	!RSX-11M/M-PLUS <u>System</u>
	! <u>Management Guide</u> .

PAR=[main\_parname:]subparname[:size]  
[NO]DIRECTORY  
[NO]OVERLAY  
[NO]VIRTUAL  
[NO]LOGICAL  
[NO]READ\_AHEAD

## QUICK REFERENCE

/NOCACHE	
/[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 -- Chapter 4

SET FILE establishes certain file attributes.

#### Format

SET FILE[/qualifier[s]] filespec[s]

#### Command Qualifiers

/END_OF_FILE: (BLOCK:n, BYTE:n)	
/ENTER:synonym_filespec	!See main text
/NOWARNINGS	!See main text
/REMOVE	!Suppresses error messages
/REWIND	!See main text
/TRUNCATE	!Magtape only; rewinds tape before !beginning operation
	!Eliminates blocks allocated but !unused; saves disk space

### SET GROUPFLAGS -- Chapter 6

SET GROUPFLAGS creates and deletes group global event flags. Nonprivileged users can set group global event flags for their group (n) only. See main text.

#### Format

SET GROUPFLAGS:n[/qualifier]

#### Command Qualifiers

/CREATE	!default
/DELETE	

### SET HOST -- Chapter 1

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 system and the remote system (nodename) must run DECnet software.

#### Format

SET HOST nodename



## QUICK REFERENCE

### SET LIBRARY/DIRECTORY -- Chapter 7

SET LIBRARY/DIRECTORY establishes the directory where the system utilities and other nonprivileged system tasks are kept. Note that this command does not create the specified directory. Privileged.

#### Format

SET LIBRARY/DIRECTORY:[directory]

### SET [NO]PARTITION -- Chapter 7

SET [NO]PARTITION creates or eliminates a partition. The partition name must be from one through six characters. Privileged.

#### Format

SET [NO]PARTITION:parname/qualifier[s]

#### Command Qualifiers

/BASE:n  
/DEVICE !Device common.  
/DIAGNOSTIC  
/SIZE:n  
/SYSTEM  
/TOP:value

### SET PASSWORD -- Chapter 3

SET PASSWORD changes your password.

#### Format

SET PASSWORD

### SET PRIORITY -- Chapter 7

SET PRIORITY alters the priority of an active task. The active task's priority (n) can be from 0 through 250. Privileged.

#### Format

SET PRIORITY:n taskname

### SET PROTECTION -- Chapter 4

SET PROTECTION changes the protection code of files. The protection code controls who can access files and in what ways. The first format is preferred. The default code is (SY:RWED,OW:RWED,GR:RWED,WO:R).

#### Format

SET PROTECTION:(code) [/qualifier[s]] filespec[s]

SET PROTECTION[/qualifier[s]] filespec[s] (code)

## QUICK REFERENCE

### 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 PROTECTION/[NO]DEFAULT -- Chapter 4

SET PROTECTION/DEFAULT establishes your personal default protection code (for example, (SY:RWED,OW:RWED,GR:R,WO:)) for all files that you create after issuing this command. It is recommended that you place this command in your LOGIN.CMD file. SET PROTECTION/NODEFAULT removes your personal default file protection.

### Format

SET PROTECTION:(code)/DEFAULT

SET PROTECTION/NODEFAULT

### SET QUEUE/ENTRY -- Chapter 4

See also the RSX-11/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. However, you can specify only one attribute at a time. See SET QUEUE/JOB to modify by job name.

### Format

SET QUEUE/ENTRY:n[/qualifier]

### Command Qualifiers

/AFTER:(dd-mmm-yy hh:mm)	
/COPIES:n	
/[NO]DELETE	
/FILE POSITION:n	
/FORMS:n	
/JOBCOUNT:n	
/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 -- Chapter 4

See also 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. However, you can specify only one attribute at a time. See SET QUEUE/ENTRY to modify by entry number.

## QUICK REFERENCE

### Format

SET QUEUE/JOB[/qualifier] queuename [[uic]]jobname

### Command Qualifiers

/AFTER:(ddd-mmm-yy hh:mm)  
/COPIES:n  
/[NO]DELETE  
/FILE\_POSITION:n  
/FORMS:n  
/JOBCOUNT:n  
/LENGTH:n  
/[NO]LOWERCASE  
/PAGE\_COUNT:n  
/PRIORITY:n

!n is 1--150 nonprivileged; through  
!250 privileged; default n=50  
!Same as RELEASE/QUEUE

/RELEASE  
/[NO]RESTART  
/[NO]UPPERCASE

### SET SYSTEM -- Chapter 8

SET SYSTEM establishes certain characteristics of the system. You must always specify a qualifier. Privileged.

### Format

SET SYSTEM/qualifier

### Command Qualifiers

/[NO]CRASH\_DEVICE[:ddn:]  
/DIRECTORY:[directory]

!Loads a crash driver.  
!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  
/PACKETS:n  
/POOL:top:max:total

!n = 0 through 15.  
!Increases size of pool; see main  
!text.

/POOL/LIMITS:arg  
HIGH=n  
LOW=n  
MINIMUM\_SIZE:n  
TASK\_PRIORITY:n

!See main text.

### SET TERMINAL -- Chapter 3

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

### Format

SET TERMINAL[:ttnn:]/qualifier[s]

# QUICK REFERENCE

## Command Qualifiers

### Group 1: Common Use

/[NO]BROADCAST	
/CLI:cliname	
/[NO]CONTROL=C	
/DCL	
/[NO]HOLD_SCREEN	!Not for VT100s
/INQUIRE	!Automatically sets proper terminal characteristics
/[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.
/[NO]DEC_CRT	
/DTC01	
/[NO]EDIT_MODE	
/[NO]FORM_FEED	
/[NO]HARDCOPY	
/[NO]HOSTSYNC	
/KSR33	
/KSR35	
/LA12	
/LA30P	
/LA30S	
/LA34	
/LA36	
/LA38	
/LA50	
/LA100	
/LA120	
/LA180S	
/LA210	
/LFFILL	
/LN03	
/LQP02	
/LQP03	
/MODEL:arg	
/PAGE_LENGTH:n	
/PRINTER_PORT	
/PRO SERIES	
/[NO]REGIS	
/[NO]SCOPE	
/[NO]SOFT_CHARACTERS	
/[NO]TAB	
/[NO]TRANSLATION_ROUTINE[:arg]	
n	!ACD number
logical	!Logical name
	!for ACD number

## QUICK REFERENCE

```
/[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  
/[NOPASTHRU  
/[NO]REMOTE  
/[NO]SERIAL                  !/SERIAL is default.  
/[NO]SLAVE  
/[NO]TYPEAHEAD[:n]          !n can be 0--255.  
/[NO]WRAP
```

### SET UIC -- Chapter 4

SET UIC changes the User Identification Code of privileged users. If your terminal's default is /NONAMED DIRECTORY, then this command also changes the directory location of a privileged user.

#### Format

```
SET UIC [uic]
```

### SHOW ACCOUNTING -- Chapter 3

SHOW ACCOUNTING displays current information on your terminal session, if you are nonprivileged. Privileged users can display information about any terminal session.

#### Format

```
SHOW ACCOUNTING/qualifier
```

#### Command Qualifiers

```
/INFORMATION  
/TRANSACTIONS[:infile] outfile
```

## QUICK REFERENCE

### SHOW ASSIGNMENTS -- Chapter 5

SHOW ASSIGNMENTS displays at your terminal your local and login logical name assignments. Privileged users can display assignments for other terminals as well as all assignments in the operating system.

SHOW LOGICAL displays the same information as SHOW ASSIGNMENTS.

#### Format

SHOW ASSIGNMENTS[/qualifier[s]]

#### Command Qualifiers

/ALL	
/GLOBAL	!Privileged
/GROUP[:g]	!UIC group number
/LOCAL	!Default
/LOGIN	!Same as /LOCAL
/SYSTEM	!Synonym for /GLOBAL; privileged
/TERMINAL:ttnn:	!Privileged

### SHOW CACHE -- RSX-11M/M-PLUS System Management Guide

SHOW CACHE displays data caching information.

#### Format

SHOW CACHE [ddnn:][[/qualifier]

#### Command Qualifiers

/RATE:n	!n is number of seconds
---------	-------------------------

### SHOW CLOCK\_QUEUE -- Chapter 7

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.

#### Format

SHOW CLOCK\_QUEUE

### SHOW COMMON -- Chapter 7

SHOW COMMON displays the names of resident commons installed in the system, their PCB addresses, the number of attached tasks, and the status of the common. If you do not specify a common name, all commons are displayed.

#### Format

SHOW COMMON[:name] [/qualifier]

#### Command Qualifier

/TASK	!Displays tasks attached to a !named common.
-------	---

## QUICK REFERENCE

### SHOW [DAY]TIME -- Chapter 8

SHOW [DAY]TIME displays the system time and date setting.

#### Format

SHOW [DAY]TIME

### SHOW DEFAULT -- Chapter 4

SHOW DEFAULT displays the current default device and directory for your terminal. This command also displays whether your terminal's default is /NAMED\_DIRECTORY or /NONAMED\_DIRECTORY, and your User Identification Code.

#### Format

SHOW DEFAULT

### SHOW DEVICES -- Chapter 5

SHOW DEVICES displays information about the devices included in the system. This command displays 2-character mnemonic names only.

#### Format

SHOW DEVICES[/qualifier]

#### Command Qualifiers

/[NO]CACHE

/dd[nn:]

/[NO]PUBLIC

/[NO]SYSTEM

!Synonym for /PUBLIC

/WIDTH:ddnn:

### SHOW HOST -- Chapter 1

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 LIBRARY -- Chapter 7

SHOW LIBRARY displays the current RSX-11M-PLUS library directory. This is the directory where the nonprivileged system utilities are kept.

#### Format

SHOW LIBRARY[/qualifier]

## QUICK REFERENCE

### Command Qualifier

/DIRECTORY

```
!Non-operational, for
!for VMS compatibility
```

## SHOW LOGICALS -- Chapter 5

SHOW LOGICALS displays at your terminal your local and login logical name assignments. Privileged users can display assignments for other terminals as well as all assignments in the operating system.

SHOW ASSIGNMENTS displays the same information as SHOW LOGICAL.

## Format

SHOW LOGICAL[/qualifier[s]]

## Command Qualifiers

/ALL

/GLOBAL

!Privileged

/ GROUP [:q]

!UIC group number

/LOCAL

!Default

/LOGIN

**! Same as /LOCAL**

/ SYSTEM

!Synonym for /GLOBAL; privileged

/TERMINAL:ttnn:

!Privileged

## SHOW MEMORY -- Chapter 8

SHOW MEMORY invokes the Resource Monitoring Display (RMD), a dynamic display of the system's activities in memory.

## Format

SHOW MEMORY

## SHOW PARTITIONS -- Chapter 7

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 -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

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



## QUICK REFERENCE

### Format

SHOW processortype processorname

### Processor Types

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 PROTECTION -- Chapter 4

SHOW PROTECTION displays your personal default file protection code. Your default file protection can be established in two ways: by issuing the SET PROTECTION/DEFAULT command and by using the Account File Maintenance Utility (ACNT) to enter a protection code for your account.

### Format

SHOW PROTECTION

### SHOW QUEUE -- Chapter 4

See also the RSX-11M/M-PLUS Batch and Queue Operations Manual.

SHOW QUEUE displays information about batch and print jobs in queues.

### Format

SHOW QUEUE[/qualifier] [queue name]

### Command Qualifiers

/ALL	!All entries in all queues. !Default.
/BATCH	!All entries in all batch 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:[uic]	!Lists only jobs from that UIC; !default UIC is login UIC.
/PRINT	!All nonbatch queues; synonym for !/DEVICE.

## QUICK REFERENCE

### SHOW SYSTEM -- Chapter 8

SHOW SYSTEM displays information about the current system.

#### Format

SHOW SYSTEM[/qualifier]

#### Command Qualifiers

/CLI	!CLIs on current system
/CRASH_DEVICE	
/DIRECTORY	!Default; displays current system !directory
/EXTENSION_LIMIT	!Task extension limit
/NETWORK_UIC	!Directory that DECnet-tasks !are located
/PACKETS	!Maximum I/O packets and number !currently available
/POOL	!Displays pool statistics
/POOL/LIMITS	!Displays pool limits
/SECONDARY_POOL	!Shows secondary pool

### SHOW TASKS -- Chapter 7

SHOW TASKS displays information about active or installed tasks.

#### Format

SHOW TASKS[:taskname]/qualifier[s]

#### Command Qualifiers

/ACTIVE[:ttnn:]	
/DEVICE:ddnn:	!Show tasks installed from named !device
/INSTALLED	
/LOGICAL_UNITS	!Static LUNs for installed task !qualifier
/BRIEF	
/FULL	
/ALL	

### SHOW TASKS/DYNAMIC -- Chapter 7

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 hardcopy terminal, SHOW TASKS/DYNAMIC provides a snapshot display. See main text for more information.

#### Format to display individual task information

SHOW TASKS:taskname/DYNAMIC[/qualifier]

#### Command Qualifier

/RATE:n	!Rate in seconds for display !change; default n=1.
---------	---

## QUICK REFERENCE

### Format to display Active Task List

SHOW TASKS/ACTIVE/DYNAMIC[/qualifier[s]]

#### Command Qualifiers

/OWNER:arg	
ddnn:	
ALL	!default
/PRIORITY:n	!Highest priority to display;
	!default n=250.
/RATE:n	!Rate in seconds for display
	!changes; default n=1.

### SHOW TERMINAL -- Chapter 3

SHOW TERMINAL displays information about your terminal and other terminals on your system. If you do not specify a qualifier, SHOW TERMINAL displays all attributes for your terminal (TI:).

#### Format

SHOW TERMINAL[:ttnn:][[/qualifier]]

#### Command Qualifiers

/[NO]ADVANCED_VIDEO	
/[NO]ANSI_CRT	
/[NO]AUTOBAUD	
/[NO]ASR33	
/[NO]ASR35	
/[NO]BLOCK_MODE	
/[NO]BROADCAST	
/CLI:cliname	
/[NO]CONTROL:C	
/[NO]CRFILL	
/DCL	
/[NO]DEC CRT	
/[NO]DTC01	
/[NO]ECHO	
/[NO]EDIT_MODE	
/[NO]EIGHT_BIT	
/[NO]ESCAPE	
/[NO]FORM_FEED	
/[NO]FULL_DUPLEX	
/[NO]HARDCOPY	
/[NO]HOLD_SCREEN	
/[NO]HOSTSYNC	
/HT	!DECnet host terminal
/[NO]INTERACTIVE	
/[NO]KSR33	
/[NO]KSR35	
/[NO]LA12	
/[NO]LA24	
/[NO]LA30P	
/[NO]LA30S	
/[NO]LA34	
/[NO]LA36	
/[NO]LA38	
/[NO]LA50	
/[NO]LA100	
/[NO]LA120	
/[NO]LA180S	

## QUICK REFERENCE

```
/[NO]LA210
/[NO]LFFILL
/[NO]LN03
/[NO]LOCAL
/LOGGED ON
/[NO]LQP02
/[NO]LQP03
/[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]REMOTE
/RT                            !DECnet host terminal

/[NO]SCOPE
/[NO]SERIAL
/[NO]SLAVE
/[NO]SOFT_CHARACTERS
/SPEED
/[NO]TAB
/TI:                          !All real terminals
/TT
/[NO]TTSYNC
/[NO]TYPE AHEAD
/[NO]UPPERCASE                !/NO UPPERCASE same as
                                !/LOWERCASE.
                                !Virtual terminal

/VT:
/[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 UIC -- Chapter 4

SHOW UIC displays your User Identification Code (UIC). Your UIC is unique and identifies you to the operating system. Your UIC also determines whether you are a privileged or nonprivileged user.

#### Format

SHOW UIC

## QUICK REFERENCE

### SHOW USERS -- Chapter 8

See also the Introduction to RSX-11M-PLUS.

SHOW USERS displays all currently logged-in terminals, including DECnet host terminals and virtual terminals, with the default directory and login UIC for each.

#### Format

SHOW USERS

### START -- Chapter 7

START resumes execution of a task stopped by a STOP\$\$ directive. The task name defaults to TTnn:

#### Format

START[/qualifier] [taskname]

#### Command Qualifier

/TERMINAL:ttnn: !Privileged

### START/processor type -- RSX-11M/M-PLUS System Management Guide

This form of the START command starts a processor, batch processor, output processor, or card-reader processor. Privileged.

#### Format

START/processor type processorname[/qualifier[s]]

#### Processortypes

APPLICATIONS PROCESSOR

BATCH PROCESSOR

CARD READER

DEVICE

INPUT

PRINTER

PROCESSOR

!INPUT is synonym.

!PRINTER is synonym.

!CARD READER is synonym.

!DEVICE is synonym.

#### 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 -- RSX-11M/M-PLUS System Management Guide

START/QUEUE starts a queue. Privileged.

#### Format

START/QUEUE queueName

### START/QUEUE/MANAGER -- RSX-11M/M-PLUS System Management Guide

START/QUEUE/MANAGER starts the Queue Manager. Privileged.

#### Format

START/QUEUE/MANAGER

### START/UNBLOCK -- Chapter 7

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[/qualifier] [taskname]

#### Command Qualifier

/TERMINAL:ttnn: !Privileged

### STOP/ABORT -- Chapter 4

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[:]

### STOP/BLOCK -- Chapter 7

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[/qualifier] [taskname]

#### Command Qualifier

/TERMINAL:ttnn: !Privileged

## QUICK REFERENCE

### STOP/processor type -- RSX-11M/M-PLUS System Management Guide

This form of the STOP command stops a processor, batch processor, card-reader processor, printer, or other output processor. Privileged.

#### Format

STOP/processor type processor name [/qualifier[s]]

#### Processortypes

APPLICATIONS PROCESSOR

BATCH PROCESSOR

CARD READER

DEVICE

INPUT

PRINTER

PROCESSOR

!INPUT is synonym.

!PRINTER is synonym.

!CARD READER is synonym.

!DEVICE is synonym.

#### Command Qualifiers

/ABORT

!See main text.

/FILE END

/JOB END

/PAUSE

### STOP/QUEUE -- RSX-11M/M-PLUS System Management Guide

STOP/QUEUE stops queues. Privileged.

#### Format

STOP/QUEUE queue name

### STOP/QUEUE/MANAGER -- RSX-11M/M-PLUS System Management Guide

STOP/QUEUE/MANAGER stops the Queue Manager. The Queue Manager stops after the current job is processed (unless you specify the /ABORT qualifier). Privileged.

#### Format

STOP/QUEUE/MANAGER [/qualifier]

#### Command Qualifier

/ABORT

!Stops QMG immediately.

### SUBMIT -- RSX-11M/M-PLUS Batch and Queue Operations Manual

SUBMIT queues QMG batch jobs consisting of one or more user batch jobs for processing by a batch processor.

#### Format

SUBMIT [/qualifier[s]] filespec[s]

## QUICK REFERENCE

### Command Qualifiers

/AFTER:TOMORROW	
/AFTER:(dd-mmm-yy hh:mm)	
/[NO]DELETE	!Deletes batch file after run; !command or filespec qualifier.
/[NO]HOLD	!Default is /NOHOLD; /HOLD has !same effect as HOLD command.
/[NO]LOG FILE	
/NAME:jobname	!1--9 characters; default is first !filename.
/[NO]PRINTER[:queueName]	!Optionally name queue for log !print job.
/PRIORITY:n	!n is 1--150 nonprivileged; through !250 privileged; default n=50.
/QUEUE:queueName	
/[NO]RESTART	
/[NO]TRANSFER	

TYPE -- Chapter 4

TYPE prints selected files on your terminal.

### Format

TYPE [/qualifier[s]] filespec[s]

### Command Qualifiers

/EXCLUDE:filespec	!Filespec can include wildcards.
/NOWARNINGS	!Suppresses error messages
/SHARED	!Allows other user 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	!From given day through given day
/TODAY	!Today only

UNFIX -- Chapter 7

UNFIX frees a fixed task or region from memory. The taskname parameter can be the name of either a task or a region. Privileged.

### Format

UNFIX[/qualifier] taskname

### Command Qualifiers

/READONLY_SEGMENT	!RO segment of multiuser task.
/REGION	



## QUICK REFERENCE

### UNLOCK -- Chapter 4

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

/EXCLUDE:filespec	!Filespec can include wildcards.
/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



## CHAPTER 10

### COMMON ERROR MESSAGES

RSX-11M-PLUS is a complex system, and error messages can stem from many sources. This chapter supplements the error messages included in many of the command descriptions in the RSX-11M-PLUS Command Language Manual.

There are two kinds of error messages described in this chapter. Most messages come from DCL or through DCL. The bulk of this chapter describes those messages. In addition, there are brief descriptions of the messages from the Task Termination Notification routine (TKTN).

#### 10.1 DCL 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.

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 to compress the files on the output disk.

## COMMON ERROR MESSAGES

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 to compress the files on the output disk.

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 inform DIGITAL of the problem. Record the command that caused the error and other information on activity at your terminal at the time.

yyy---Bad use of wildcards in destination file name

**Explanation:** A wildcard (\*) was specified for an output file where it is not permitted.

**User Action:** Reenter the command with a complete and explicit file specification for the output file.

yyy---Cannot find directory file

**Explanation:** The command specified a directory not found on the current volume.

**User Action:** Reenter the command after checking for the correct directory and correct volume.

yyy---Cannot find files

**Explanation:** The file or files specified in the command are not in the designated directory.

**User Action:** Check the file specification and reenter the command line.

## COMMON ERROR MESSAGES

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.

yyy---Close failure on input file

or

yyy---Close failure on output file

**Explanation:** A file named in the command could not be properly closed. The file is locked.

**User Action:** Use UNLOCK to unlock the file. Determine the cause of the error and correct it if you can. 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.

## COMMON ERROR MESSAGES

### 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 for proper syntax and reenter command.

### 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.

### 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.

## COMMON ERROR MESSAGES

### 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 commands. 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 next section. Determine the cause of the error and correct it if you can. If you cannot, see your system manager or operator.

### 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.

## COMMON ERROR MESSAGES

yyy---Failed to delete file

or

yyy---Failed to mark file for delete

**Explanation:** The command attempted to delete a protected file.

**User Action:** Check for the proper file specification and default directory and reenter the command. You may need privilege to delete the file.

yyy---Failed to enter new filename

**Explanation:** The command specified a file that already exists in the directory, or the directory is protected.

**User Action:** Check for 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.

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 report the problem to DIGITAL. 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 directory 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.



## COMMON ERROR MESSAGES

### 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 directory 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 directory 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.

### yyy---File is lost

**Explanation:** The system has removed a file from its directory, 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.

## COMMON ERROR MESSAGES

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.

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.

## COMMON ERROR MESSAGES

### yyy---Illegal filespec

**Explanation:** The command required a file specification that was not present.

**User Action:** Check command for proper syntax and reenter. See Chapter 4 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 logical

**Explanation:** A specified logical name either does not exist or is not able to be translated.

**User Action:** Check command to be sure that you specified the correct logical name, or issue a SHOW ASSIGNMENTS command to see what the logical name translates into. If the logical name needs to be changed, deassign it, then reassign it to a new equivalent name. Retry command.

### 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 Chapter 4 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---Illegal user default protection code

**Explanation:** You did not specify all four user categories when issuing SET PROTECTION/DEFAULT the first time during a terminal session.

**User Action:** Reenter the command line, specifying all four user categories in the protection code.

### yyy---Improper or illegal logical

**Explanation:** You specified a defined logical name that DCL could not process (for example, a logical name that has a circular translation).

**User Action:** Check the logical name definitions (using the SHOW ASSIGNMENTS), correct the definition, and reenter the command line.

## COMMON ERROR MESSAGES

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.

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.

## COMMON ERROR MESSAGES

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 directory.

**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 five conditions exists:

1. The device is not on line.
2. The device is not mounted.
3. The hardware has failed.
4. The output volume is full.
5. 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.

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 for proper syntax and reenter the command.

yyy---Key specification out of sequence

**Explanation:** The command included improper syntax in the key definition argument of the /KEY qualifier.

**User Action:** Check for proper syntax and reenter the command.

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.

## COMMON ERROR MESSAGES

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 protection specified for any field.

**Explanation:** You attempted to set your default file protection code to deny file access to all four user categories---that is, a file protection code that looks like (SYSTEM,OWNER,GROUP,WORLD).

**User Action:** The minimum default file protection code is READ access for one user category. Reenter the command line, specifying at least the minimum file access.

yyy---No such file

**Explanation:** The command requested operations on a file that does not exist.

**User Action:** Make sure you have named the file properly. Check your defaults to be sure you are looking in the right directory on the right device. You may have made a typing error. If the desired file is in fact not present, find out why it is not present and proceed accordingly.

yyy---No user default protection specified

**Explanation:** You do not have a personal default file protection code.

**Explanation:** You can use either the SET PROTECTION/DEFAULT command or the Account File Maintenance Utility (ACNT) to establish your own default file protection. You must be privileged to use ACNT.

yyy---Not a directory device

**Explanation:** A directory-oriented command named a device that does not have directories, such as a line printer.

**User Action:** If you can, reenter the command without specifying a directory.

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.

## COMMON ERROR MESSAGES

yyy---Numeral expected

or

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.

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 directory to be sure that the file is present, not locked, and in the proper format. If the file is locked, it may be corrupted or contain bad data. Determine the cause of the error and correct it if you can. 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 the command.



## COMMON ERROR MESSAGES

### 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.

### 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 is required.

**User Action:** Reenter the command using proper syntax. The Radix-50 characters are the uppercase alphabet, the numerals 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.

## COMMON ERROR MESSAGES

### 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:** Check for proper syntax and reenter the command.

### 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:** Check for proper syntax and reenter the command.

### 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.

### 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.

## COMMON ERROR MESSAGES

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 /KEY qualifier.

**User Action:** Check for proper syntax and reenter the command.

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.

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.

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.

## COMMON ERROR MESSAGES

### 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.

## 10.2 TASK TERMINATION MESSAGES

The following messages are issued by the Task Termination Notification (TKTN) routine. 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 the messages (along with any other terminal output) in case anyone asks to see them.

### 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.

### 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.

## COMMON ERROR MESSAGES

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.

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.

## COMMON ERROR MESSAGES

### 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.

**APPENDIX A**  
**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 MCR commands are indicated by the uppercase letters (for example, ALLocate indicates the MCR command ALL). The following table gives these relationships, with any necessary comments.

Table A-1  
Functions Initiated By DCL Commands

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	ASN	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	

(continued on next page)

# FUNCTIONS INITIATED BY DCL COMMANDS

Table A-1 (Cont.)  
Functions Initiated By DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
COBOL	COB	
CONTINUE	RESume	
CONVERT	RMS CNV	
COPY	PIP	
CORAL	COR	
CREATE	PIP, FCS	
CREATE/DIRECTORY	UFD	
DEALLOCATE	DEAllocate	
DEASSIGN	ASN	Deassign logical names made with ASSIGN
	DFL	Deassign logical names made with ASSIGN
DEASSIGN/QUEUE	QMG	Eliminate assign- ment of queue to processor
DEBUG	DEBug	
DEFINE	DFL	
DELETE	PIP	Delete files
DELETE/DIRECTORY	UFD	
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	

(continued on next page)



# FUNCTIONS INITIATED BY DCL COMMANDS

Table A-1 (Cont.)  
Functions Initiated By DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
EDIT	EDT, EDI, KED, SLP	
FORTTRAN	FOR, F4P, F77	
FIX	FIX	
HELP	HELlo	As part of HELlo, HELP can run on unlogged-on terminals.
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	
MACRO	MAC	
MCR	MCR	
MOUNT	MOUnT	
PRINT	QMG	
PURGE	PIP	
RELEASE/ENTRY	QMG	
RELEASE/JOB	QMG	
REMOVE	REMOve	

(continued on next page)

# FUNCTIONS INITIATED BY DCL COMMANDS

Table A-1 (Cont.)  
Functions Initiated By DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
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	NFT	
SET LIBRARY/DIRECTORY	SET /LIBUIC	
SET PARTITION	PAR, SET /MAIN, SET /PAR, SET SUB	
SET PASSWORD	ACNT	
SET PRIORITY	ALTer	
SET PROTECTION	PIP	
SET PROTECTION/DEFAULT	PIP DEF/PR	
SET QUEUE	QMG	
SET SYSTEM/DIRECTORY	SET /SYSUIC	
SET SYSTEM/EXTENSION_LIMIT	SET /MAXEXT	
SET SYSTEM/NOLOGINS	SET /NOLOGON	
SET SYSTEM/PACKETS	SET /MAXPKT	
SET SYSTEM/POOL	SET /POOL	
SET TERMINAL	SET /attribute	
SET UIC	SET /UIC	
SHOW ACCOUNTING	ACC	Resource Accounting
SHOW ASSIGNMENTS	ASN	
SHOW CACHE	RMD C	
SHOW CLOCK_QUEUE	CLQ	

(continued on next page)

# FUNCTIONS INITIATED BY DCL COMMANDS

Table A-1 (Cont.)  
Functions Initiated By DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
SHOW COMMON	CBD	Common Block Directory
SHOW [DAY]TIME	TIM	
SHOW DEFAULT	SET /UIC ASN	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 LIBRARY/DIRECTORY	SET /LIBUIC	
SHOW LOGICAL	ASN	
SHOW MEMORY	RMD	Resource Monitoring Display
SHOW PARTITIONS	PAR	
SHOW PROCESSOR	QMG	
SHOW PROTECTION	SET /DPRO	
SHOW QUEUE	QMG	
SHOW SYSTEM/CLI	CLI /SHOW	
SHOW SYSTEM/DIRECTORY	SET /SYSUIC	
SHOW SYSTEM/EXTENSION_LIMIT	SET /MAXEXT	
SHOW SYSTEM/POOL	SET /POOL	
SHOW SYSTEM/SECONDARY_POOL	SET /SECPOL	
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	

(continued on next page)

# FUNCTIONS INITIATED BY DCL COMMANDS

Table A-1 (Cont.)  
Functions Initiated By DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
SHOW TERMINAL	DEV	
SHOW TERMINAL/attribute	SET /attribute	
SHOW UIC		
SHOW USERS	DEV	
SORT	SRT	SORT-11
START	UNS	
START/PROCESSOR	QMG	
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	
SUBMIT	QMG	
TYPE	PIP	
UNFIX	UNFix	
UNLOCK	PIP	

# INDEX

- Abbreviation (DCL), 1-12, 9-1
  - caution, 1-23
- ABORT command, 7-17 to 7-20, 9-1
- Aborting tasks
  - CTRL/C, 3-20
- Account File Maintenance Program
  - See ACNT
- ACNT, 3-10 to 3-12, 4-2
- ACP, 5-4, 5-53
- Active task, 7-1 to 7-2
- ALLOCATE command, 5-11, 5-45 to 5-47, 9-2
- Allocated device, 5-10 to 5-11
- ANALYZE command
  - ANALYZE/CRASH\_DUMP, 9-2 to 9-3
  - ANALYZE/MEDIA, 9-3
- ANALYZE/MEDIA command, 5-11, 5-19 to 5-25
- Ancillary Control Processor
  - See ACP
- ANSI file name, 1-15
- APPEND command, 4-65 to 4-67, 9-4
- ASSIGN command, 5-16, 5-26 to 5-33, 9-4
  - ASSIGN/QUEUE, 9-4
  - ASSIGN/REDIRECT, 5-34 to 5-36, 9-5
  - ASSIGN/TASK, 7-44 to 7-45, 9-5
- BACK SPACE key, 3-14
- Backup and Restore Utility
  - See BACKUP command
- BACKUP command, 5-90 to 5-128, 9-5 to 9-6
  - backing up
    - Files-11, 5-90
    - foreign volume, 5-90
  - command line, 5-91
  - error messages, 5-103 to 5-128
  - examples, 5-101 to 5-103
  - qualifiers, 5-91
    - disk processing, 5-100
    - display, 5-99
    - initialization, 5-95 to 5-98
    - selective backup and restore, 5-93 to 5-95
    - tape control, 5-98 to 5-99
    - verification, 5-99
  - restoring
    - Files-11, 5-90
    - foreign, 5-90
- Bad block, 5-11, 5-19 to 5-25, 5-74
- Batch job
  - stopping, 4-96
- Batch processing, 1-21
- Beep, 3-24
- Bell, 3-24
- Blank disk, 5-11 to 5-13
- BROADCAST command, 3-32 to 3-35, 9-7
- BRU (Backup and Restore Utility)
  - See BACKUP command
- Buzz, 3-24
- CANCEL command, 7-23, 9-7
- CAPS LOCK key, 3-15
- CL: pseudo device, 5-6, 5-7
- CLI, 1-1
- Clock queue, 7-13 to 7-16, 7-23, 9-36
  - deleting entry, 7-23
  - displaying, 7-73
- CO: pseudo device, 5-6, 5-7
- Colon (:), 1-14
- Command file, 6-6
- Command line
  - length, 3-13
- Command Line Interpreter
  - See CLI
- Comments in command lines, 1-18
- Common
  - displaying, 7-74 to 7-75
  - status flags, 7-74 to 7-75
- CONTINUE command, 7-21 to 7-22, 9-7
- Control key
  - See CTRL key
- /CONTROL qualifier
  - SET TERMINAL command, 3-40
- CONVERT command, 4-55 to 4-61, 9-7
- COPY command, 4-47 to 4-51, 9-8
- COPY key, 3-16
- Crash, 9-2 to 9-3
- CREATE command, 4-15 to 4-16, 9-9
  - CREATE/DIRECTORY, 4-17 to 4-19, 9-9
- CTRL key, 3-20
  - CTRL/C, 3-20, 7-18 to 7-20
  - CTRL/G, 3-24
  - CTRL/I, 3-24

# INDEX

## CTRL key (Cont.)

- CTRL/K, 3-24
- CTRL/L, 3-23
- CTRL/M, 3-24
- CTRL/O, 3-21
- CTRL/Q, 3-21
- CTRL/R, 3-23
- CTRL/S, 3-21
- CTRL/U, 3-22
- CTRL/X, 3-23
- CTRL/Z, 3-22

## Date, 1-13

### Date-related qualifier (DCL), 4-9

#### DCL, 1-1, 9-1

- command, 1-12
- command line, 1-7 to 1-8
- correcting errors, 1-20 to 1-21
- device and volume control
  - commands, 1-5
- file control commands, 1-4
- functional grouping, 1-2
- program development commands, 1-6
- RSX-11M-PLUS and VAX/VMS, 1-23
- SET and SHOW commands, 1-3
- system control commands, 1-7
- task control commands, 1-6

#### DCL error message, 10-1 to 10-18

#### DEALLOCATE command, 5-48 to 5-49, 9-9

#### DEASSIGN command, 5-37 to 5-40, 9-9

#### DEASSIGN/QUEUE, 9-10

#### DEBUG command, 7-46 to 7-47, 9-10

#### Decimal number, 1-13

#### DECnet software, 1-16

#### Default

- file specification, 4-4 to 4-6

#### DEFINE command, 5-14, 5-41 to 5-44, 9-10

#### DELETE command, 4-40 to 4-43, 9-11

#### DELETE/DIRECTORY, 4-20, 9-11

#### DELETE/ENTRY, 4-86 to 4-87, 9-11

#### DELETE/JOB, 4-86 to 4-87, 9-12

#### DELETE/PROCESSOR, 9-12

#### DELETE/QUEUE, 9-12

#### DELETE key, 3-14

#### Device

- allocated, 5-10 to 5-11
- displaying attributes, 5-135

## Device (Cont.)

- logical, 5-9 to 5-10, 5-26 to 5-33, 5-37 to 5-40, 5-129 to 5-131

- assignment, 5-9

- logical name, 5-9 to 5-10

- setting attributes, 5-132

- unowned, 5-10 to 5-11

#### DIFFERENCES command, 4-68 to 4-73, 9-12

#### DIGITAL Command Language

- See DCL

#### Directory, 4-1, 4-3

- creating, 4-17

- default, 4-4

- set, 4-100

- show, 4-104

- deleting, 4-20

- library, 7-28 to 7-32, 8-10

- displaying, 8-12

- setting, 8-11

- list, 4-31

- maintenance, 4-30

- named, 4-100

- numbered, 4-100

- protection, 4-13, 4-18, 4-111 to 4-114

- default, 4-115 to 4-117

- system, 7-28 to 7-32, 8-10

#### DIRECTORY command, 4-31 to 4-39, 9-13

#### DISMOUNT command, 5-12, 5-13, 5-68 to 5-72, 9-14

#### Displaying time, 8-4

#### EDI (Line Text Editor), 4-28

#### EDIT command, 4-21 to 4-30

- EDIT/EDI, 4-28 to 4-30, 9-15

- EDIT/EDT, 4-22 to 4-25, 9-14

- EDIT/SLP, 4-26 to 4-28, 9-14

- /RECOVER qualifier, 4-23

#### EDT Editor, 4-22

- EDIT command, 9-14

#### ENTER key, 3-14

#### Entry number

- QMG, 4-86

#### Equal sign (=)

- DCL, 1-14

#### Error

- correcting, 1-20

#### Error message

- BACKUP command, 5-103 to 5-128

- DCL, 10-1 to 10-18

- INITIALIZE command, 5-82 to 5-85

# INDEX

- Error message (Cont.)
  - INSTALL command, 7-33 to 7-38
  - LIBRARY command, 6-57 to 6-62
  - MOUNT command, 5-64 to 5-67
- ESCAPE key, 3-14
- Event flag
  - displaying, 6-65
  - setting, 6-63 to 6-64
- /EXCLUDE qualifier (DCL), 4-9
- File
  - appending, 4-65
  - attributes
    - setting, 4-105
  - comparing, 4-68
  - copying, 4-47
  - creating, 4-15, 4-21
  - deleting, 4-40
  - editing, 4-21
    - recovering, 4-23
  - indirect command
    - See ICP
  - list, 4-31
  - location, 4-2
  - owner, 4-2
  - printing, 4-76
  - protection, 4-10 to 4-12, 9-31
    - default
      - setting, 4-115 to 4-117
      - showing, 4-118
      - setting, 4-111 to 4-114
  - purging, 4-44
  - renaming, 4-52
  - source
    - editing, 4-26
    - typing, 4-62
    - unlocking, 4-74
    - version number, 4-3
      - default, 4-4
- File name, 4-3
- File organization, 4-1
- File specification, 4-2 to 4-4
  - defaults, 4-4 to 4-6
  - format, 4-2
  - wildcard, 4-6 to 4-8
- File type, 4-3
  - default, 4-4, 4-5 to 4-6
- Files-11, 5-4
  - volume
    - backing up, 5-90
    - initializing, 5-73
    - restoring, 5-90
- FIX command, 7-41 to 7-42, 9-15
- Flag page, 4-78 to 4-79
- /FOREIGN qualifier
  - MOUNT command, 5-4, 5-10, 5-11, 5-13, 5-53
- Form feed, 3-23
- HELP
  - getting, 3-25 to 3-28
- HELP command, 1-10 to 1-12, 3-25 to 3-28, 9-16
- Help file, 3-28 to 3-31
  - writing, 3-28
- HOLD command
  - HOLD/ENTRY, 4-97, 9-16
  - HOLD/JOB, 4-97, 9-16
- HOLD SCREEN key, 3-16
- ICP, 1-22
- INITIALIZE command, 5-4, 5-11, 5-13, 5-73 to 5-85, 9-16
  - error messages, 5-82 to 5-85
- INITIALIZE/PROCESSOR
  - input, 9-17
  - output, 9-18
- INITIALIZE/QUEUE, 9-18
- INITIALIZE/UPDATE, 5-86 to 5-89, 9-18
- /INQUIRE qualifier
  - SET TERMINAL command, 3-41
- INSTALL command, 7-28 to 7-38, 9-19
  - error messages, 7-33 to 7-38
- Installed task, 7-1 to 7-2, 7-3, 7-12 to 7-16
- Installing task, 7-28 to 7-38
- Keyboard, 3-13 to 3-18
  - keys table, 3-16 to 3-18
- VT200-series terminal, 3-1
- Keypad, 3-18 to 3-19
  - alternate, 3-18 to 3-19
  - editing
    - VT100-series terminal, 3-18
    - VT200-series terminal, 3-19
- Laboratory device, 5-5
- LB: pseudo device, 5-6, 5-7
- Library, 6-38 to 6-40
  - macro, 6-40
  - object module, 6-39
  - universal, 6-40
- LIBRARY command
  - LIBRARY/REMOVE, 6-54
- LIBRARY command, 6-38 to 6-62, 9-19
  - error messages, 6-57 to 6-62

# INDEX

- LIBRARY command (Cont.)
  - LIBRARY/COMPRESS, 6-40 to 6-41, 9-20
  - LIBRARY/CREATE, 6-42 to 6-44, 9-20
  - LIBRARY/DELETE, 6-45 to 6-46, 9-20
  - LIBRARY/EXTRACT, 6-46 to 6-47, 9-20
  - LIBRARY/INSERT, 6-47 to 6-49, 9-21
  - LIBRARY/LIST, 6-50 to 6-54, 9-21
  - LIBRARY/REMOVE, 6-55, 9-21
  - LIBRARY/REPLACE, 6-55 to 6-56, 9-21
  - /SQUEEZE qualifier, 6-44
- Library directory, 7-28 to 7-32
  - displaying, 8-12
  - setting, 8-11
- Library files, 6-4 to 6-5
- Line continuation, 1-18
- Line terminators, 3-14
- LINK command, 6-10 to 6-37, 9-22 to 9-23
  - description, 6-15
  - examples, 6-34 to 6-36
  - introduction, 6-10 to 6-15
  - option, 6-31
  - /OVERLAY DESCRIPTION qualifier, 6-6, 6-23
  - qualifier, 6-16 to 6-30
- Logical device, 5-6, 5-9 to 5-10
  - assigning, 5-26 to 5-33
  - assignment, 5-9
  - deleting, 5-37 to 5-40
  - displaying, 5-129 to 5-131
- Logical device name, 5-9 to 5-10
- Logical name, 5-6
  - assigning, 5-26 to 5-33
  - defining, 5-41 to 5-44
  - deleting, 5-37 to 5-40
  - description, 5-14 to 5-18
  - displaying, 5-15, 5-129 to 5-131
  - format, 5-16
  - tables, 5-15
  - translation, 5-17
- Logical name table, 5-15
- Logical unit number
  - See LUN
- LOGIN command, 3-4 to 3-8, 9-23
  - password, 3-4
  - UIC, 3-4
- LOGOUT command, 3-9 to 3-10, 5-14, 7-20, 9-23
- LUN, 5-6, 5-8 to 5-9, 7-44 to 7-45, 7-68 to 7-70
- Macro library, 6-40
- Magnetic tape
  - blank, 5-13
  - mounting, 5-62 to 5-67
  - new, 5-13
  - preparing for use, 5-13
  - volume
    - labeled, 1-15
- Magtape
  - See Magnetic tape
- Master File Directory
  - See MFD
- MCR, 1-1 to 1-2
  - command, 1-12
- MCR command, 9-24
- MFD, 4-1, 4-2, 5-12
- MK: pseudo device, 5-6, 5-7
- Monitor Console Routine
  - See MCR
- MOUNT command, 5-12, 5-13, 5-50 to 5-67, 9-24 to 9-25
  - error messages, 5-64 to 5-67
  - qualifiers
    - /FOREIGN, 5-4, 5-10, 5-11, 5-13, 5-53
    - /NOLABEL, 5-4, 5-58
    - /NOSHAREABLE, 5-55
    - /SHAREABLE, 5-14, 5-55
- Mounted volume, 5-10 to 5-11
- Multiple parameters, 1-14
- /NAMED DIRECTORY qualifier
  - SET DEFAULT command, 4-100 to 4-101
- Network, 1-16
- New disk, 5-11 to 5-13
- NL: pseudo device, 5-7
- NO SCROLL key, 3-16
- /NOCONTROL qualifier
  - SET TERMINAL command, 3-40
- Node, 1-16
- /NOLABEL qualifier
  - MOUNT command, 5-4, 5-58
- /NONAMED DIRECTORY qualifier
  - SET DEFAULT command, 4-100 to 4-101
- /NOSERIAL qualifier
  - SET TERMINAL command, 3-49
- /NOSHAREABLE qualifier
  - MOUNT command, 5-55



# INDEX

- Number, 1-13
- Object file, 6-1, 6-3, 6-4, 6-10
- Object module library, 6-39
- Octal number, 1-13
- .ODL file type, 6-5
- Overlay files, 6-5 to 6-6
- /OVERLAY DESCRIPTION qualifier
  - LINK command, 6-6, 6-23
- Parameter
  - displaying system, 8-8 to 8-10
  - setting system, 8-5 to 8-7
- Partition
  - memory
    - displaying, 7-52 to 7-54
    - setting, 7-48 to 7-51
- Password, 3-4
  - changing, 3-11
- Peripheral device, 5-1 to 5-11
- Pool
  - displaying, 8-15 to 8-18
- PRINT command, 4-76 to 4-85, 9-25
- Print job
  - stopping, 4-96
- Printer
  - stopping, 4-96
- Priority
  - task, 7-71 to 7-72
- Private device, 5-45 to 5-49
- Private volume, 5-10 to 5-11, 5-45 to 5-49, 5-55
- Program development, 6-1 to 6-37
- Prompt, 1-8
- Protection, 4-9 to 4-14, 4-111 to 4-118, 9-31
  - directory, 4-13, 4-18
  - file, 4-10 to 4-12, 9-31
  - default
    - setting, 4-115 to 4-117
    - showing, 4-118
    - setting, 4-111 to 4-114
    - specifying, 5-97
  - volume, 4-13 to 4-14
  - specifying, 5-76
- Pseudo device, 3-2, 5-1, 5-6 to 5-8
  - CL:, 5-6, 5-7
  - CO:, 5-6, 5-7
  - LB:, 5-6, 5-7
  - MK:, 5-6, 5-7
  - NL:, 5-7
  - RD:, 5-7
  - SP:, 5-7
  - SY:, 5-7
- Pseudo device (Cont.)
  - TI:, 5-7, 5-8
  - VT:, 5-7, 5-8
  - WK:, 5-7, 5-8
- Public device, 5-14
- Public volume, 5-10 to 5-11, 5-55
- PURGE command, 4-44 to 4-46, 9-26
- QMG, 1-21, 4-85 to 4-98
  - entry number, 4-86
- Qualifier, 1-9
- Queue Manager
  - See QMG
- Quoted string, 1-15
- RD: pseudo device, 5-7
- Record
  - moving, 4-55
- /RECOVER qualifier
  - EDIT command, 4-23
- RELEASE command
  - RELEASE/ENTRY, 4-98, 9-26
  - RELEASE/JOB, 4-98, 9-27
- REMOVE command, 7-39 to 7-40, 9-27
- RENAME command, 4-52 to 4-54, 9-27
- REPEAT key, 3-15
- REQUEST command, 3-36 to 3-37, 9-27
- Resource Accounting, 3-9, 3-12 to 3-13, 9-35
- RETURN key, 3-13 to 3-14, 3-24
- RSX-11M-PLUS Operating System, 1-1
- RUN command, 7-5 to 7-16, 9-28
  - installed task, 7-12 to 7-16
  - introduction, 7-5 to 7-7
  - uninstalled task, 7-8 to 7-12
- Scratch disk
  - formatting, 5-11
  - preparing for use, 5-11 to 5-13
- /SERIAL qualifier
  - SET TERMINAL command, 3-49
- SET command
  - SET [DAY]TIME, 8-2, 9-29
  - SET DEBUG, 1-19, 9-29
  - SET DEFAULT, 4-100 to 4-103, 9-29
  - /NAMED DIRECTORY qualifier, 4-100 to 4-101
  - /NONAMED DIRECTORY qualifier, 4-100 to 4-101

# INDEX

## SET command (Cont.)

- SET DEVICE, 5-132 to 5-134, 9-29
- SET FILE, 4-105 to 4-107, 9-30
- SET GROUPFLAGS, 6-63 to 6-64, 9-30
- SET HOST, 1-16, 9-30
- SET LIBRARY/DIRECTORY, 8-11, 9-31
- SET PARTITION, 7-48 to 7-51, 9-31
- SET PASSWORD, 3-11 to 3-12, 9-31
- SET PRIORITY, 7-71 to 7-72, 9-31
- SET PROTECTION, 4-111 to 4-114, 9-31
- SET PROTECTION/DEFAULT, 4-115 to 4-117, 9-32
- SET QUEUE, 4-92 to 4-95
- SET QUEUE/ENTRY, 9-32
- SET QUEUE/JOB, 9-32
- SET SYSTEM, 8-5 to 8-7, 9-33
- SET UIC, 4-108 to 4-109, 9-35
- SET TERMINAL command, 3-20, 3-38 to 3-52, 9-33 to 9-35
  - qualifiers
    - /CONTROL, 3-40
    - /INQUIRE, 3-41
    - /NOCONTROL, 3-40
    - /NOSERIAL, 3-49
    - /SERIAL, 3-49
- Setting time, 8-2
- /SHAREABLE qualifier
  - MOUNT command, 5-14, 5-55
- Shareable volume, 5-10 to 5-11, 5-55
- SHIFT key, 3-15
- SHOW command
  - SHOW [DAY]TIME, 9-37
  - SHOW [DAY]TIMESort="D", 8-4
  - SHOW ACCOUNTING, 9-35
  - SHOW ASSIGNMENTS, 5-15, 5-129 to 5-131, 9-36
  - SHOW CACHE, 9-36
  - SHOW CLOCK\_QUEUE, 7-73, 9-36
  - SHOW COMMON, 7-74 to 7-75, 9-36
  - SHOW DEFAULT, 4-104, 9-37
  - SHOW DEVICES, 5-5, 5-135 to 5-138, 9-37
  - SHOW GROUPFLAGS, 6-65
  - SHOW HOST, 1-17, 9-37
  - SHOW LIBRARY, 8-12, 9-37
  - SHOW LOGICALS, 5-15, 5-129 to 5-131, 9-38

## SHOW command (Cont.)

- SHOW MEMORY, 8-15 to 8-18, 9-38
- SHOW PARTITIONS, 7-52 to 7-54, 9-38
- SHOW PROCESSOR, 4-91, 9-38
- SHOW PROTECTION, 4-118, 9-39
- SHOW QUEUE, 4-88 to 4-90, 9-39
- SHOW SYSTEM, 8-8 to 8-10, 9-40
- SHOW TASKS, 7-55 to 7-70, 9-40
- SHOW TASKS/ACTIVE, 7-55 to 7-60
- SHOW TASKS/DYNAMIC, 7-64 to 7-68, 9-40
- SHOW TASKS/INSTALLED, 7-60 to 7-63
- SHOW TASKS/INSTALLED/DEVICE, 7-63 to 7-64
- SHOW TASKS/LOGICAL\_UNITS, 7-68 to 7-70
- SHOW TERMINAL, 3-53 to 3-61, 9-41 to 9-42
- SHOW UIC, 4-110, 9-42
- SHOW USERS, 8-13 to 8-14, 9-43
- SLP, 4-26
- Source file, 6-1, 6-3
- Source Language Input Program
  - See SLP
- SP: pseudo device, 5-7
- Special keys
  - VT100-series terminal, 3-19
  - VT200-series terminal, 3-19
  - VT52 terminal, 3-19
- /SQUEEZE qualifier
  - LIBRARY command, 6-44
- START command, 7-26, 9-43
  - START/PROCESSOR, 9-43
  - START/QUEUE, 9-44
  - START/QUEUE/MANAGER, 9-44
  - START/UNBLOCK, 7-27, 9-44
- Status flags
  - common, 7-74 to 7-75
  - task, 7-57 to 7-59
- STD, 7-1 to 7-5
- STOP command
  - STOP/ABORT, 4-96, 9-44
  - STOP/BLOCK, 7-24 to 7-25, 9-44
  - STOP/PROCESSOR, 9-45
  - STOP/QUEUE, 9-45
  - STOP/QUEUE/MANAGER, 9-45
- String
  - quoted, 1-15
- SUBMIT command, 9-45
- SY: pseudo device, 5-7
- Symbol
  - TKB, 6-8

## INDEX

- System
  - displaying parameters, 8-8 to 8-10
  - setting parameters, 8-5 to 8-7
- System directory, 7-28 to 7-32, 8-10
- System status
  - displaying, 8-15 to 8-18
- System Task Directory
  - See STD
- System users
  - displaying, 8-13 to 8-14
- Tab, 3-24
  - vertical, 3-24
- Task, 7-1
  - aborting, 7-17 to 7-20
  - active, 7-1 to 7-2
    - displaying, 7-55 to 7-60
    - dynamic display, 7-64 to 7-68
  - blocking, 7-24 to 7-25
  - continuing, 7-21 to 7-22
  - dormant, 7-1
  - executing, 7-5 to 7-16
  - fixing, 7-41 to 7-42
  - installation, 7-28 to 7-38
  - installed, 7-1 to 7-2, 7-3, 7-12 to 7-16
    - displaying, 7-60 to 7-63
  - priority
    - setting, 7-71 to 7-72
  - removing, 7-39 to 7-40
  - starting, 7-26
  - unblocking, 7-27
- Task build
  - See TKB
- Task image file (.TSK), 6-1, 6-3, 6-10, 7-8 to 7-12
- Task name, 7-2 to 7-5
- Task status flags, 7-57 to 7-59
- Task termination message, 10-18 to 10-20
- Task Termination Notification routine
  - See TKTN
- Terminal, 3-1 to 3-13
  - attached, 3-2
  - characteristic
    - setting, 3-38
    - showing, 3-53
  - characteristics
    - setting, 9-33
    - showing, 9-41
  - slaved, 3-2
  - VT100-series, 3-18
- Terminal
  - VT100-series (Cont.)
    - special keys, 3-19
  - VT200-series, 3-19
    - special keys, 3-19
  - VT52, 3-19
    - special keys, 3-19
- Terminal Operations commands, 1-3
- TI: pseudo device, 3-2, 5-7, 5-8
- Time
  - displaying, 8-4
  - setting, 8-2
- TKB, 6-1, 6-6 to 6-9, 9-22 to 9-23
  - input file, 6-3 to 6-6
    - command, 6-6
    - library, 6-4 to 6-5
    - object, 6-4, 6-10
    - overlay, 6-5 to 6-6
  - output file
    - task image, 6-10
  - symbol, 6-8
  - task creation, 6-7
- TKTN message, 10-18 to 10-20
- TYPE command, 4-62 to 4-64, 9-46
- UFD
  - See Directory
- UIC, 4-2, 4-108 to 4-110
- Underscore ( \_ ), 1-14
- UNFIX command, 7-43, 9-46
- Universal library, 6-40
- UNLOCK command, 4-74 to 4-75, 9-47
- Unowned device, 5-10 to 5-11
- User File Directory (UFD)
  - See Directory
- User Identification Code
  - See UIC
- Utilities, 1-22
- VAX/VMS DCL, 1-23
- Version number
  - file, 4-3
  - default, 4-4
- Vertical tab, 3-24
- Volume, 4-1, 5-1 to 5-11
  - Files-11, 5-4
  - mounted, 5-10 to 5-11
  - mounting, 5-50 to 5-67
  - private, 5-10 to 5-11, 5-45 to 5-49
  - protection, 4-13 to 4-14
  - public, 5-10 to 5-11
  - shareable, 5-10 to 5-11

## INDEX

VT100-series terminal, 3-18  
VT200-series terminal, 3-19  
    keyboard, 3-1  
VT52 terminal, 3-19

VT: pseudo device, 5-7, 5-8  
Wildcard, 4-6 to 4-8  
WK: pseudo device, 5-7, 5-8

### 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 \_\_\_\_\_

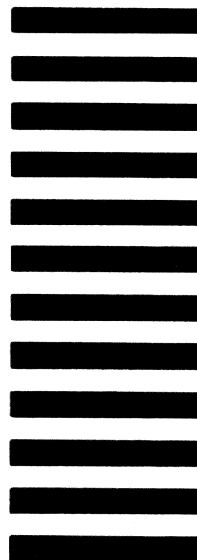
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or Country

- — 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 — -

Cut Along Dotted Line