RSX–11M–PLUS Command Language Manual

Order No. AA–FD04B–TC

RSX–11M–PLUS Version 4.0

Digital Equipment Corporation Maynard, Massachusetts

First Printing, October 1979
Revised, November 1981
Revised, April 1983
Revised, June 1985
Revised, September 1987

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, 1987 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation:

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

ZK3319

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA & PUERTO RICO*	CANADA	INTERNATIONAL
Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario K2K 2A6 Attn: Direct Order Desk	Digital Equipment Corporation PSG Business Manager c/o Digital's local subsidiary or approved distributor

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.

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

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a trademark of the American Mathematical Society.

Contents

Preface xv

Summary of Technical Changes xxi

Chapter 1 How to Use DCL

1.1 The RSX-11M-PLUS Operating System 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 1-8
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 Quoting Strings 1-14
1.3.10 ANSI File Names 1-15
1.3.11 DECnet and DCL 1-15
1.3.11.1 SET HOST 1-17
1.3.11.2 SHOW HOST 1-19
1.3.12 Command Line Continuation 1-20
1.3.13 Comments in Command Lines 1-20

1.3.14 SET DEBUG 1-20
1.3.15 Typing Errors 1-22
1.4 Using the Queue Manager: Print and Batch Jobs 1-23
1.5 Indirect Command Files in DCL 1-23
1.6 DCL and RSX-11M-PLUS Utilities and System Tasks 1-23
1.7 RSX-11M-PLUS DCL and VMS DCL 1-24

Chapter 2 The SET and SHOW Commands

2.1 SET and SHOW TIME 2-2
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 2-5
2.6 SHOW ASSIGNMENTS and SHOW LOGICALS 2-5
2.7 SET and SHOW DEFAULTS 2-6
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 2-9
2.12 SET DEBUG 2-10

Chapter 3 Terminal Operations

3.1 Logging In and Logging Out 3-2
3.1.1 LOGIN 3-4
3.1.2 LOGOUT 3-8
3.1.3 The Account File Maintenance Program (ACNT) 3-9
3.1.3.1 SET PASSWORD 3-10
3.1.3.2 Resource Accounting on RSX-11M-PLUS 3-11
3.2 The Keyboard 3-12
3.2.1 RETURN Key and Command Line Length 3-12
3.2.2 Line Terminators 3-13
3.2.3 DELETE Versus BACK SPACE 3-13
3.2.4 CAPS LOCK and SHIFT 3-14
3.2.5 REPEAT 3-14
3.2.6 NO SCROLL or HOLD SCREEN 3-15
3.2.7 Other Keys 3-15
3.3 Keypads 3-17
3.4 The CTRL Key and Control Characters 3-18
3.4.1 CTRL/C 3-19

3.4.2 CTRL/S and CTRL/Q . . . 3-19
3.4.3 CTRL/O . . . 3-20
3.4.4 CTRL/Z . . . 3-20
3.4.5 CTRL/U . . . 3-20
3.4.6 CTRL/R . . . 3-21
3.4.7 CTRL/X . . . 3-21
3.4.8 CTRL/L . . . 3-21
3.4.9 Other Control Characters . . . 3-22
3.5 HELP . . . 3-23
3.5.1 The HELP Facility . . . 3-26
3.5.1.1 Writing Help Files . . . 3-27
3.6 BROADCAST . . . 3-32
3.7 REQUEST . . . 3-36
3.8 SET and SHOW TERMINAL . . . 3-38
3.8.1 SET TERMINAL . . . 3-38
3.8.2 SHOW TERMINAL . . . 3-51

Chapter 4 Handling Files

4.1 File Ownership and Location . . . 4-1
4.2 File Specification . . . 4-2
4.3 Defaults in File Specifications . . . 4-4
4.4 Wildcard Characters in File Specifications . . . 4-6
4.5 Date-Related Qualifiers . . . 4-8
4.6 Protection . . . 4-9
4.6.1 File Protection . . . 4-9
4.6.2 Directory Protection . . . 4-12
4.6.3 Volume Protection . . . 4-13
4.7 Creating Directories and Sequential Files . . . 4-14
4.7.1 CREATE . . . 4-14
4.7.2 CREATE/DIRECTORY . . . 4-16
4.7.3 DELETE/DIRECTORY . . . 4-19
4.7.4 EDIT . . . 4-20
4.7.4.1 EDIT[/EDT] . . . 4-20
4.7.4.2 EDIT/SLP . . . 4-24
4.7.4.3 EDIT/EDI and EDIT/editor . . . 4-26
4.8 Maintaining a Directory . . . 4-29
4.8.1 DIRECTORY . . . 4-29
4.8.2 DELETE . . . 4-38
4.8.3 PURGE . . . 4-42
4.8.4 COPY . . . 4-45
4.8.5 RENAME . . . 4-50

4.8.6 CONVERT 4-53
4.9 Other File-Related Commands 4-59
4.9.1 TYPE 4-60
4.9.2 APPEND 4-63
4.9.3 DIFFERENCES 4-65
4.9.4 UNLOCK 4-71
4.10 PRINT and Queue Manager Commands 4-73
4.10.1 PRINT 4-73
4.10.2 DELETE 4-83
4.10.3 SHOW QUEUE 4-85
4.10.4 SHOW PROCESSOR 4-88
4.10.5 SET QUEUE 4-89
4.10.6 STOP/ABORT 4-92
4.10.7 HOLD 4-93
4.10.8 RELEASE 4-94
4.11 SET and SHOW 4-95
4.11.1 SET DEFAULT 4-96
4.11.2 SHOW DEFAULT 4-100
4.11.3 SET FILE 4-101
4.11.4 SET UIC 4-104
4.11.5 SHOW UIC 4-106
4.11.6 SET PROTECTION 4-107
4.11.7 SET PROTECTION/[NO]DEFAULT 4-110
4.11.8 SHOW PROTECTION 4-113

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, Logical Unit Numbers, 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 Commands Related to Devices and Volumes 5-13
5.2 Logical Names 5-14
5.2.1 Logical Name Tables 5-14
5.2.1.1 Displaying Logical Name Table Entries 5-15
5.2.2 How to Create and Delete Logical Names 5-15

5.2.3 Logical Name Translation 5-16
5.2.3.1 Iterative Translation 5-16
5.3 ANALYZE/MEDIA 5-18
5.4 ASSIGN 5-24
5.4.1 For RSX-11M-PLUS Systems Without Extended Logical Name Support 5-24
5.4.2 For RSX-11M-PLUS Systems With Extended Logical Name Support 5-26
5.4.3 ASSIGN/REDIRECT 5-31
5.5 DEASSIGN 5-34
5.6 DEFINE 5-37
5.7 ALLOCATE 5-41
5.8 DEALLOCATE 5-44
5.9 MOUNT 5-46
5.9.1 Mounting Magnetic Tapes and Tape Sets 5-58
5.10 DISMOUNT 5-63
5.11 INITIALIZE 5-67
5.11.1 INITIALIZE/FORMAT 5-80
5.11.2 INITIALIZE/UPDATE 5-87
5.12 BACKUP 5-91
5.12.1 Backup Operations 5-91
5.12.2 Restore Operations 5-92
5.12.3 Disk-to-Disk Copy Operations 5-92
5.12.4 Mounting Disk and Tape Devices for Backup Operations 5-92
5.12.5 The Backup Command Line 5-93
5.12.5.1 BACKUP Command Qualifiers 5-94
5.12.5.2 Selective Backup and Restore Qualifiers 5-96
5.12.5.3 Initialization Qualifiers 5-97
5.12.5.4 Tape Control Qualifiers 5-98
5.12.5.5 Verification Qualifiers 5-100
5.12.5.6 Display Qualifiers 5-100
5.12.5.7 Disk Processing Qualifiers 5-101
5.12.6 Examples 5-103
5.12.7 BACKUP Error Messages 5-105
5.12.7.1 Informational Messages 5-106
5.12.7.2 Warning Messages 5-107
5.12.7.3 Fatal Messages 5-118
5.13 SET and SHOW 5-131
5.13.1 SHOW ASSIGNMENTS and SHOW LOGICALS 5-131
5.13.2 SET DEVICE 5-134
5.13.3 SHOW DEVICES 5-138
5.13.4 SHOW CACHE 5-142

Chapter 6 The LINK and LIBRARY Commands

6.1 Linking a Task with the Task Builder 6-1
6.2 Review of the Programming Process 6-1
6.2.1 Basic Steps in Programming 6-2
6.2.2 The Source File 6-2
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 6-3
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 6-5
6.4 The Task Builder Functions 6-6
6.4.1 Creating a Task 6-6
6.4.2 Arranging Addresses 6-7
6.4.3 Finding Symbols and Their Definitions 6-7
6.5 The LINK Command 6-7
6.5.1 LINK 6-13
6.6 LIBRARY 6-32
6.6.1 LIBRARY/COMPRESS 6-35
6.6.2 LIBRARY/CREATE 6-37
6.6.3 LIBRARY/DELETE 6-40
6.6.4 LIBRARY/EXTRACT 6-41
6.6.5 LIBRARY/INSERT 6-43
6.6.6 LIBRARY/LIST 6-45
6.6.7 LIBRARY/REMOVE 6-49
6.6.8 LIBRARY/REPLACE 6-50
6.6.9 Error Messages from the LIBRARY Command 6-52

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 7-6
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 7-37
7.10 FIX 7-39
7.11 UNFIX 7-41
7.12 ASSIGN/TASK 7-42
7.13 DEBUG 7-44
7.14 SET and SHOW 7-46
7.14.1 SET PARTITION 7-46
7.14.2 SHOW PARTITIONS 7-50
7.14.3 SHOW TASKS 7-53
7.14.3.1 SHOW TASKS/ACTIVE 7-54
7.14.3.2 SHOW TASKS/INSTALLED 7-60
7.14.3.3 SHOW TASKS/INSTALLED/DEVICE 7-62
7.14.3.4 SHOW TASKS/ACTIVE/DYNAMIC 7-63
7.14.3.5 SHOW TASKS:taskname/DYNAMIC 7-66
7.14.3.6 SHOW TASKS/LOGICAL_UNITS 7-68
7.14.4 SET PRIORITY 7-70
7.14.5 SHOW CLOCK_QUEUE 7-72
7.14.6 SHOW COMMON 7-73
7.15 SET and SHOW 7-75
7.15.1 SET GROUPFLAGS 7-75
7.15.2 SHOW GROUPFLAGS 7-77

Chapter 8 System Control

8.1 SET AND SHOW 8-1
8.1.1 SET [DAY]TIME 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 8-10
8.2.2 SHOW LIBRARY 8-11
8.3 SHOW USERS 8-12
8.4 SHOW MEMORY 8-14

Chapter 9 Quick Reference

ABORT 9-2
ALLOCATE 9-2
ANALYZE/CRASH_DUMP 9-2
ANALYZE/MEDIA 9-3
APPEND 9-4
ASSIGN 9-4
ASSIGN/QUEUE 9-5
ASSIGN/REDIRECT 9-5
ASSIGN/TASK 9-5
BACKUP 9-5
BROADCAST 9-7
CANCEL 9-7
CONTINUE 9-8
CONVERT 9-8
COPY 9-9
CREATE 9-9
CREATE/DIRECTORY 9-10
DEALLOCATE 9-10
DEASSIGN 9-10
DEASSIGN/QUEUE 9-11
DEBUG 9-11
DEFINE 9-11
DELETE 9-12
DELETE/DIRECTORY 9-12
DELETE/ENTRY 9-12
DELETE/JOB 9-13
DELETE/processortype 9-13
DELETE/QUEUE 9-13
DIFFERENCES 9-14
DIRECTORY 9-14
DISMOUNT 9-15
EDIT[/EDT] 9-15
EDIT/SLP 9-16
EDIT/EDI and EDIT/editor 9-16
FIX 9-17
HELP 9-17
HOLD/ENTRY 9-17
HOLD/JOB 9-18
INITIALIZE 9-18
INITIALIZE/processortype (Input) 9-19

INITIALIZE/processortype (Output) 9-19
INITIALIZE/FORMAT 9-20
INITIALIZE/QUEUE 9-20
INITIALIZE/UPDATE 9-20
INSTALL 9-21
LIBRARY 9-22
LIBRARY/COMPRESS 9-22
LIBRARY/CREATE 9-22
LIBRARY/DELETE 9-23
LIBRARY/EXTRACT 9-23
LIBRARY/INSERT 9-23
LIBRARY/LIST 9-23
LIBRARY/REMOVE 9-24
LIBRARY/REPLACE 9-24
LINK 9-24
LOGIN 9-26
LOGOUT 9-26
MCR 9-26
MOUNT 9-26
PRINT 9-28
PURGE 9-29
RELEASE/ENTRY 9-30
RELEASE/JOB 9-30
REMOVE 9-30
RENAME 9-30
REQUEST 9-31
RUN Installed Task 9-31
RUN Uninstalled Task 9-31
SET [DAY]TIME 9-32
SET DEBUG 9-32
SET DEFAULT 9-32
SET DEVICE 9-33
SET FILE 9-33
SET GROUPFLAGS 9-34
SET HOST 9-34
SET LIBRARY/DIRECTORY 9-34
SET [NO]PARTITION 9-34
SET PASSWORD 9-34
SET PRIORITY 9-35
SET PROTECTION 9-35
SET PROTECTION/[NO]DEFAULT 9-35
SET QUEUE/ENTRY 9-36

SET QUEUE/JOB 9-36
SET SYSTEM 9-37
SET TERMINAL 9-37
SET UIC 9-39
SHOW ACCOUNTING 9-40
SHOW ASSIGNMENTS 9-40
SHOW CACHE 9-40
SHOW CLOCK_QUEUE 9-40
SHOW COMMON 9-41
SHOW [DAY]TIME 9-41
SHOW DEFAULT 9-41
SHOW DEVICES 9-41
SHOW HOST 9-42
SHOW LIBRARY 9-42
SHOW LOGICALS 9-42
SHOW MEMORY 9-43
SHOW PARTITIONS 9-43
SHOW PROCESSOR 9-43
SHOW PROTECTION 9-43
SHOW QUEUE 9-44
SHOW SYSTEM 9-44
SHOW TASKS 9-45
SHOW TASKS/ACTIVE/DYNAMIC 9-45
SHOW TERMINAL 9-46
SHOW UIC 9-48
SHOW USERS 9-48
START 9-48
START/processortype 9-48
START/QUEUE 9-49
START/QUEUE/MANAGER 9-49
START/UNBLOCK 9-49
STOP/ABORT 9-49
STOP/BLOCK 9-50
STOP/processortype 9-50
STOP/QUEUE 9-50
STOP/QUEUE/MANAGER 9-51
SUBMIT 9-51
TYPE 9-51
UNFIX 9-52
UNLOCK 9-52

Chapter 10 Common Error Messages

10.1 DCL Error Messages 10-1
10.2 Task Termination Messages 10-17

Appendix A Functions Initiated by DCL Commands

Index

Figures

3–1 VT200-Series Keyboard and Keypad 3-2
3–2 Help Files for the SET TERMINAL Command 3-31
4–1 File Organization on a Volume 4-2
4–2 Job Flag Page 4-76
4–3 File Flag Page 4-77
6–1 Module Address Assignment 6-8

Tables

3–1 Keyboard and Keypad Characters 3-15
3–2 Display of Terminal Attributes from SHOW TERMINAL 3-53
4–1 File Types 4-5
5–1 Devices on RSX–11M–PLUS Systems 5-2
5–2 Pseudo Devices on RSX–11M–PLUS Operating Systems 5-7
5–3 Use of /MOUNTED and /INITIALIZE Qualifiers with the BACKUP Command 5-93
6–1 Task Builder Options 6-26
7–1 Base and Size Specifications for SET PARTITION 7-47
7–2 Task Status Flags 7-56
A–1 Functions Initiated by DCL Commands A-1

Preface

Manual Objectives

The *RSX-11M-PLUS Command Language Manual* provides information needed to use an RSX-11M-PLUS operating system. Important system concepts are introduced and explained, including use of 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 all users of RSX-11M-PLUS systems. The manual is a reference manual with many tutorial elements, but new users should read the *Introduction to RSX-11M-PLUS* first.

Structure of This Document

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 Introduces the basic concepts of DCL and includes many hints on using DCL.

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

Chapter 3 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 Includes more detail about commands that list directories and that create, delete, rename, copy, type, and print files. Wildcard characters and defaults are explained.

Chapter 5	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	Documents the program-development facilities included in RSX–11M–PLUS. The chapter explains the Task Builder and the LIBRARY command, which is used to maintain libraries.
Chapter 7	Explains how tasks run in the system, and how they are named, installed, fixed in memory, and controlled while running. In addition, the chapter describes commands that abort tasks, place tasks in the clock queue, and display information about tasks in the system.
Chapter 8	Explains system-control commands including those which set the time and date, set and show certain system characteristics, and show memory utilization.
Chapter 9	Lists alphabetically 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	Lists and explains error messages common to several DCL commands. Other command-specific error messages are explained in the full command descriptions.
Appendix A	Lists the functions of major DCL commands.

Associated Documents

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–PLUS MCR Operations Manual* and the *RSX–11M–PLUS Utilities Manual.*

Programmers should see the *RSX–11M–PLUS Guide to Program Development.*

Conventions Used in This Document

The following conventions are used in this manual:

Convention	Meaning
$	A dollar sign followed by a space is the default prompt of the DIGITAL Command Language (DCL), which is one of the command interfaces used on RSX–11M–PLUS and Micro/RSX systems. Many systems include DCL.
MCR>	This is the explicit prompt of the Monitor Console Routine (MCR).
DCL>	This is the explicit prompt of the DIGITAL Command Language (DCL).
xxx>	Three characters followed by a right angle bracket indicate the explicit prompt for a task, utility, or program on the system.
UPPERCASE	Uppercase letters in a command line indicate letters that must be entered as they are shown. For example, utility switches must always be entered as they are shown in format specifications.
command abbreviations	Where short forms of commands are allowed, the shortest form acceptable is represented by uppercase letters. The following example shows the shortest abbreviation allowed for the DCL command DIRECTORY: `$ DIR`
lowercase	Any word in lowercase must be substituted for. Usually the lowercase word identifies the kind of substitution expected, such as the word *filespec*, which indicates that you should fill in a file specification.
/keyword, /qualifier, or /switch	A command element preceded by a slash (/) is an MCR keyword; a DCL qualifier; or a task, utility, or program switch. Keywords, qualifiers, and switches alter the action of the command they follow.
parameter	Required command fields are generally called parameters. The most common parameters are file specifications.
[option]	Square brackets indicate optional entries in a command line. If the brackets include syntactical elements, such as periods (.) or slashes (/), those elements are required for the field. If the field appears in lowercase, you are to substitute a valid command element if you include the field. Note that when an option is entered, the brackets are not included in the command line.

Convention	Meaning
[, . . .]	Square brackets around a comma and an ellipsis mark indicate that you can use a series of optional elements separated by commas. For example, (argument[, . . .]) means that you can specify a series of optional arguments by enclosing the arguments in parentheses and by separating them with commas.
{ }	Braces indicate a choice of required options. You are to choose from one of the options listed.
: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). In DCL, the equal sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.
( )	Parentheses are used to enclose more than one argument in a command line. `SET PROT = (S:RWED,O:RWED)`
,	Commas are used as separators for command line parameters and to indicate positional entries on a command line. Positional entries are those elements that must be in a certain place in the command line. Although you might omit elements that come before the desired element, the commas that separate them must still be included.
[g,m] [directory]	The convention [g,m] signifies a User Identification Code (UIC). The g is a group number and the m is a member number. The UIC identifies a user and is used mainly for controlling access to files and privileged system functions. This may also signify a User File Directory (UFD), commonly called a directory. A directory is the location of files. Other notations for directories are: [ggg,mmm], [gggmmm], [ufd], [name], and [directory]. The convention [directory] signifies a directory. Most directories have 1- to 9-character names, but some are in the same [g,m] form as the UIC.
filespec	A full file specification includes device, directory, file name, file type, and version number, as shown in the following example: `DL2:[46,63]INDIRECT.TXT;3` Full file specifications are rarely needed. If you do not provide a version number, the highest numbered version is used. If you do not provide a directory, the default directory is used. Some system functions default to particular file types. Many commands accept a wildcard character (*) in place of the file name, file type, or version number. Some commands accept a filespec with a DECnet node name.

Convention	Meaning
.	A period in a file specification separates the file name and file type. When the file type is not specified, the period may be omitted from the file specification.
;	A semicolon in a file specification separates the file type from the file version. If the version is not specified, the semicolon may be omitted from the file specification.
@	The at sign invokes an indirect command file. The at sign immediately precedes the file specification for the indirect command file, as follows: `@filename[.filetype;version]`
. . .	A horizontal ellipsis indicates the following: • Additional, optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis shows where elements of command input or statements in an example or figure have been omitted because they are irrelevant to the point being discussed.
KEYNAME	This typeface denotes one of the keys on the terminal keyboard; for example, the RETURN key.
"print" and "type"	Generally, the term "print" refers to any output sent to a terminal by the system. The term "type" refers to any user input from a terminal. In DCL, the PRINT command requests a file to be printed on a printer. The TYPE command requests a file to be printed (displayed) on the terminal.
black ink	In examples, what the system displays on the terminal is printed in black.
red ink	In interactive examples, what the user types is printed in red.

Convention	Meaning
[xxx]	A symbol with a 1- to 3-character abbreviation, such as [x] or [RET], indicates that you press a key on the terminal. For example, [RET] indicates the RETURN key, [LF] indicates the LINE FEED key, and [DEL] indicates the DELETE key.
[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 some terminals as ^Z. However, not all control characters echo.

See Chapter 1 for more information on DCL conventions.

Summary of Technical Changes

Version 4.0 of the RSX–11M–PLUS operating system includes two new DIGITAL Command Language (DCL) commands and several new command qualifiers. It also supports several new devices.

New Commands

DCL includes the following new commands:

- INITIALIZE/FORMAT[qualifier[s]]

  Command Qualifiers

  /BADBLOCKS
  /DENSITY:arg
  HIGH or DOUBLE
  LOW or SINGLE
  /ERROR_LIMIT:n
  /MANUAL
  /[NO]MESSAGE
  /OVERRIDE
  /[NO]VERIFY
  /WRITE_LAST_TRACK:n

- CLEAR HOST

The INITIALIZE/FORMAT command formats and verifies volumes on disk cartridges, disk packs, and fixed media disks associated with any RSX–11M–PLUS operating system that includes online formatting support in the Executive.

The CLEAR HOST command breaks the connection to the remote node that has been established with the SET HOST command.

New Qualifiers

DCL includes the following new command qualifiers:

BACKUP[qualifier[s]]
/IDENTIFICATION

INSTALL[qualifier[s]]
/[NO]DEFER_BINDING

LINK[qualifier[s]]
/CODE:OTS_FAST

MOUNT[qualifier[s]]
/CACHE:[NO]DEFER_WRITES

PRINT[qualifier[s]]
/[NO]ADJACENT

SET DEVICE[qualifier[s]]
/CACHE:[NO]DEFER_WRITES

SET TERMINAL[qualifier[s]]
/CHARACTER_LENGTH:n

SHOW TERMINAL[qualifier[s]]
/CHARACTER_LENGTH:n

New Hardware Support

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

Terminal Types	Disk Drives	Magnetic Tape Drives
LA75	RA82	TK70
	RD31	TU81
	RD32	TU81-E
	RD54	TUK50
	RX33	

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 are interested in a quick review of DCL and the operating system, you may want to read the *Introduction to RSX-11M-PLUS*. This book includes an overview of the operating system, including examples of all of the most commonly used DCL commands.

DCL commands are full words that describe the actions to be taken, not abbreviations or mnemonics. Thus, the command that sets a terminal to lowercase is SET TERMINAL /LOWERCASE.

You are not required to use the full form of DCL commands at the terminal, however. Usually, you need to type only the command elements required to form a unique command. For instance, the abbreviation SET TERM/LOW will 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 information, see the individual command descriptions.

1.1 The RSX-11M-PLUS Operating System 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 system. MCR also provides commands that set and display certain system characteristics. MCR provides the most direct interface with the RSX–11M–PLUS operating system.

In general, MCR commands must be entered in exact syntax. Most MCR commands are abbreviations or mnemonics.

DCL is an optional user-oriented CLI included in most systems with many users. Commands in DCL are English or English-like words and follow well-defined syntax rules. DCL is consistent and designed for 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 RSX–11M–PLUS, RSX–11M, and VMS. 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 1.3.14 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 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 chapter contains an introduction to the important concepts of the functional group and fully detailed descriptions of each command.

Command descriptions can have five parts, as follows:

- *A brief statement of the function* of the command.
- *A complete description of the format* of the command, including prompts, defaults, acceptable values for arguments, and a list of qualifiers with their effects.
- *Examples* of the command in use. These examples often illustrate less obvious aspects of the use of the command.
- *Notes*, including warnings about side effects, counteracting commands, recommendations for further reading, and so forth. Some command descriptions do not have notes.
- *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.

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. 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 described at the end of each chapter except Chapter 6. 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.

1.2.2 Terminal Operations

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

These commands enable you to perform the following operations:

Operation	DCL Command
Log you in to the computer	LOGIN
Log you out of the computer	LOGOUT
Invoke online 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

In addition, Chapter 3 includes a detailed description of the use of the terminal keyboard on RSX-11M-PLUS systems. All control characters (CTRL/x) 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
Stop a current print or batch job from completing	STOP/ABORT

Operation	DCL Command
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

Operation	DCL Command
Disable a mounted volume	DISMOUNT
Produce a volume in Files–11 format	INITIALIZE
Back up 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.

1.2.6 Running Tasks

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

The SET GROUPFLAGS and SHOW GROUPFLAGS commands are included at the end of this chapter.

These commands enable you to perform 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

Operation	DCL Command
Load and lock a task into memory	FIX
Unlock a fixed task in memory	UNFIX
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 equal 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 files you want listed. Most DCL parameters are file specifications, as is this one. The asterisk (*) indicates a wildcard character file type. See Section 4.4 for information on wildcard characters.

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 ALTMODE. 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 want to continue with the command, press CTRL/Z.

1.3.2 Qualifiers

Qualifiers modify the action of the command and are generally optional. Qualifiers are always preceded by 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 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 a hyphen (-) 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 invoke 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
```

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:

```
$ HELP TYPE TODAY RET

 TYPE/TODAY filespec[,s]

 The /TODAY qualifier specifies that you want 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.

While being prompted by a command, you can also get help on a specific subtopic 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 "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.

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 command:

```
$ SET TERMINAL/VT100 [RET]
```

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

```
$ SET VT100 [RET]
```

to achieve the same result. This second form does not work on VMS systems.

These briefer forms should be used interactively only, and not when you are making a permanent record or creating an indirect command file (or batch file). 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
DEALLOCATE	DEAL	MACRO	M
DEASSIGN	DEAS	PRINT	P
EDIT	E	RUN	R
FORTRAN	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, because DCL distinguishes between the two. In rare cases, the command description directs you to add a decimal point to identify a decimal number.

DCL recognizes dates in two forms, as follows:

```
dd-mmm-yy as in 21-JUN-87
```

or

```
mm/dd/yy as in 06/21/87
```

System displays are always in the first format.

1.3.7 Multiple Parameters

Some commands permit you to enter more than one parameter. 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 when 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 within a command.

1.3.9 Quoting Strings

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

```
$ BROADCAST/ALL Rock and roll will never die [RET]
```

is broadcast as

```
11-MAY-87 13:55          From PRINCE::WRITERS (TT64) to ALL
ROCK AND ROLL WILL NEVER DIE
```

while

```
$ BROADCAST/ALL "Rock and roll will never die" [RET]
```

is broadcast as

```
11-MAY-87 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 the MOUNT command or the /COMMAND qualifier to the RUN or INSTALL commands.

1.3.10 ANSI File Names

ANSI file names do not have the same format as Files–11 file names and may contain characters that are invalid in Files–11. For instance, the file name "MONEY&." is a valid ANSI file name, as is "CASH&.". To use these 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 to 9, the letters of the alphabet, the dollar sign ($), and the period (.).

1.3.11 DECnet and DCL

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

DECnet is a DIGITAL product that enables two or more systems to communicate with 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. 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 follows:

```
$ 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.

1.3.11.1 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

Parameter

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, your terminal session is returned to your local node. You can break the connection with the remote node by pressing CTRL/ \ (the control key and a backslash) and then pressing RETURN. The local node responds with a prompt, and you can then enter the CLEAR HOST command to return your terminal session to your local node. For an example, see the Examples section following. If you change your mind about returning your terminal session to your local node, you can enter the CONTINUE command after pressing CTRL/ \ and RETURN. Your terminal session is then continued on the remote 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 to 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.

Examples

```
$ SET HOST TOOTSI [RET]

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

SET HOST

```
$ LOGIN [RET]
Account or Name: Erin [RET]
Password:
    .
    .
    .
$
```

This SET HOST command connects the user to the remote node TOOTSI. The remote system identifies itself and prompts for a user name. The user then logs in to the remote system.

```
$ SET HOST POLLUX [RET]
Connected to "POLLUX", System type = RSX-11M-PLUS
System ID:  RSX TIMESHARING
$ LOGIN [RET]
Account or name:JONES
Password:
LOG -- Invalid account
$ LOGIN [RET]
Account or name: [CTRL/ \]  [RET]
MOON> CLEAR HOST [RET]
NCT -- Control returned to node MOON
$
```

This SET HOST command connects the user to the remote node POLLUX. However, the user does not have an account on this node, so the node rejects the user's name and password and returns the dollar sign prompt. At this point, the user logs in again and presses CTRL/ \ (the control key and a backslash) and the RETURN key. The local system (MOON) responds with the prompt **MOON>** . The user then enters the CLEAR HOST command and presses RETURN to return the terminal session to the local node.

1.3.11.2 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.

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 V4.0 BL40
```

This example indicates the display from SHOW HOST. The name of your current processor is TOOTSI, which is running Version 4.0 of the RSX-11M-PLUS operating system.

1.3.12 Command Line Continuation

Sometimes a command will not fit on one line. In that case you can continue the line by ending the command line with a hyphen (-) and pressing RETURN. The DCL continuation prompt (->) then 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.

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. Each blank space, 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-87) 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.

1.3.13 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. Here is another example:

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

1.3.14 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-PLUS and Micro/RSX 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-87
$ 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 are 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.

1.3.15 Typing 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.

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 by users or by system tasks or applications tasks. 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 processing allows you to place commands in a file and execute them as often as you wish and at any time you wish without retyping them. This means jobs that require a great deal of CPU time can be run when there are fewer demands on the system; for example, at night or on weekends.

All uses of the Queue Manager and the batch processor are described in the *RSX–11M–PLUS Batch and Queue Operations Manual*. That manual includes sample print and batch jobs and full descriptions of Queue Manager and batch processor commands, including the commands used to display jobs in queues, place jobs on queues, and alter the status of jobs in queues. For information on setting up and running the Queue Manager, consult the *RSX–11M–PLUS and Micro/RSX System Management Guide*.

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 repeatedly, you can place these commands in a file to be run by Indirect.

Indirect accepts not only DCL and MCR commands but also directives. Indirect is described in full in the *RSX–11M–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 it 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–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.

1.7 RSX–11M–PLUS DCL and VMS DCL

DIGITAL Command Language (DCL) on RSX–11M–PLUS Version 3.0 and subsequent releases is designed to be compatible with DCL on 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 VMS system. In most cases, the default forms of DCL commands will have the same effect on either 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, 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 are rarely used.

SET and SHOW both issue the following 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 page numbers.

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 }

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-87 or 6/4/87. In the latter case, mm stands for month number. No matter how you enter the date, it will be displayed in the first form.

See Section 8.1.1.

The 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 Resource Monitoring Display (RMD), which provides information on task activity, partitions, drivers, and CPU usage on your system.

Format

SHOW MEMORY

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

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

Not all systems include this command for interactive use. However, most systems 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

You use this command in the following manner:

```
$ SHOW TERMINAL [RET]
TT22:     [7,77]   [7,77]    27-FEB-87 11:29      1       J. KAULLMAN
          CLI   = MCR     BUF   = 255.    HFILL = 0       SPEED=(9600:9600)
          LINES = 24.     TERM  = VT100   OWNER = none    NOPARITY
          CHAR_LENGTH = 8 NOPRINTER_PORT  NOPASTHRU       NOSERIAL
          LOWER   NOPRIV  NOHOLD  SLAVE   NOESC   CRT     NOFORM  NOREMOTE
          ECHO    NOVFILL HHT     NOFDX   WRAP    NORPA   NOEBC   TYPEAHEAD
          NOCTRLC AVO     ANSI    DEC     NOEDIT  NOREGIS NOSOFT  NOBLKMOD
          HSYNC   BRO     NOABAUD TTSYNC
```

The display is explained in full in Section 3.8.2. The following is a brief explanation of some of the fields:

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 (CTRL/C) 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. For example, a command such as the following is accepted by the system and executed:

```
$ SET TERMINAL/VT100/LOWERCASE/DCL [RET]
```

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	ASR35	DTC01	KSR33
LA12	LA30P	LA30S	LA34
LA36	LA38	LA50	LA75
LA100	LA120	LA180S	LA200_SERIES
LA210	LN03	LQP01	LQP02
LQP03	PRO_SERIES	VT05B	VT50
VT52	VT55	VT61	VT100
VT101	VT102	VT105	VT125
VT131	VT132	VT200_SERIES	

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

2.4 SHOW DEVICES

This 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. The argument 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 (QMG). The following command displays information about all jobs in all queues:

Format

SHOW QUEUE

In addition to showing all queue assignments, the display includes all Queue Manager jobs and all files included in those jobs. This display includes batch queues as well as print queues.

The following command displays full information about all jobs in all queues:

Format

SHOW QUEUE/FULL

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-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]]

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 the *RSX-11M-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 LP1: is controlled by processor LP1:.

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

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 letters A to Z or the numbers 0 to 9. A numbered directory is two octal numbers from 1 to 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 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 want to protect more than one file, you must separate the file specifications by commas or use a wildcard character (*).

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.

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 want 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 want 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 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. This chapter tells you how to use a terminal to communicate with the RSX–11M–PLUS operating system. For a quick review of basic operations at a terminal, consult the *Introduction to RSX–11M–PLUS.*

The discussions in this chapter refer to four DIGITAL terminals: the LA36 hardcopy terminal and the VT100-, VT200-, and VT300-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. Also, the manual shipped with your terminal will provide complete information on the use of your terminal on the system. In general, what is true of the LA36 is true of many models of hardcopy terminal, and what is true of the VT100 is 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 have some sort of keyboard; some terminals do not have a keypad.

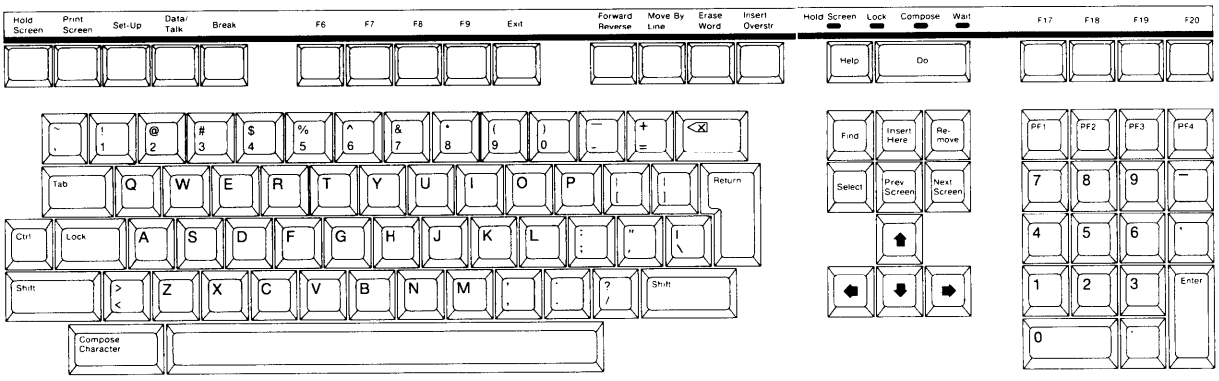
Each terminal on an RSX–11M–PLUS system has a number. Your terminal number is displayed when you log in. Your terminal number is also used by the system to identify tasks run from your terminal. (See Chapter 7 for more information on naming tasks.)

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 is printed on your terminal.

Figure 3–1: VT200-Series Keyboard and Keypad



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 attaches your terminal.

Terminals may also be reserved for a particular 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 at login. 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 finished 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 RET
```

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, because 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. To log out, you type the following command:

```
$ 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 vice versa) 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 Sections 3.1.1 and 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), as follows:

```
[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 characters, including dollar signs ($), periods (.), exclamation points (!), single 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 UIC, the password does print. Your system manager establishes your password in setting up your account. You can change your password with the SET PASSWORD command (see Section 3.1.3.1).

Examples

```
$ LOGIN [RET]
Account or name: KAFKA [RET]
Password:        [RET]

RSX-11M-PLUS V4.0 BL40 [3,54] System POLLUX
20-MAY-87  09:16  Logged on Terminal TT54:

Good Morning

May 20, 1987            System Manager

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

           ** BACKUPS TONIGHT **

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

May 1, 1987         Operator

Magtape drive MU0: 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 V4.0 BL40 [3,54] System POLLUX
20-MAY-87  09:16  Logged on Terminal TT54:

Good Morning

May 20, 1987
System Manager

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

           ** BACKUPS TONIGHT **

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

$
```

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

```
$ LOGIN 303/5 [RET]
Password:       [RET]

RSX-11M-PLUS V4.0 BL40 [3,54] System POLLUX
20-MAY-87  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 V4.0 BL40 [3,54] System POLLUX
20-MAY-87  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: DU1:
$ @ <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-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.

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-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, probably because someone else is logging in or using the ACNT task.

User Action: 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.

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 USERS to find out which user has your terminal allocated.

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]

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 in again 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-87 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 RET
DM0 -- 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-87 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-87  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 LO to log out.

BYE is the same as LOGOUT.

3.1.3 The Account File Maintenance Program (ACNT)

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. 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–PLUS and Micro/RSX System Management Guide*. Nonprivileged users have no access to ACNT. They can change their password using the SET PASSWORD command (see the next section). However, to change other aspects of their accounts, nonprivileged users need to see the system manager.

The *RSX–11M–PLUS and Micro/RSX System Management Guide* also has information on Resource Accounting, a separate system program that keeps track of system usage.

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 characters, including dollar signs ($), periods (.), exclamation points (!), single quotation marks ('), and hyphens (-).

Example

```
$ SET PASSWORD [RET]
Old password: <KRYPTO> [RET]
New password: <COVERS> [RET]
Verification: <COVERS> [RET]
$
```

In this example, the user changes the password from KRYPTO to COVERS. 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 systems, 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.

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 invalid.

User Action: Type in a password consisting of 39 or fewer valid characters. The valid characters are the uppercase letters A to Z, the numbers 0 to 9, the period (.), the dollar sign ($), the exclamation point (!), the single 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 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.

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-PLUS and Micro/RSX System Management Guide.*

3.2 The Keyboard

Most of the keys on the main keyboard are self-explanatory. They function much the same as the keys 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.

3.2.1 RETURN Key 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
```

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-87
$
```

You need not wait for the return of the prompt to enter another command unless serial mode is activated. For more information on serial mode, see the description of the /SERIAL qualifier to the SET TERMINAL command in Section 3.8.1.

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-87
```

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 <X] on the VT200-series terminals) works in a slightly different fashion on video and hardcopy terminals.

On hardcopy terminals, the DELETE key removes the last character typed and then verifies the removal by repeating the removed character on your terminal, preceded by a backslash. This action is repeated each time you press the DELETE key. For instance, the 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 CTRL/R after the corrected line to retype it properly, without the intervening deleted characters, thus:

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

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.

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 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	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.
		At monitor level, the control key sometimes echoes as a circumflex when struck in combination with a letter.
		If you want to include control characters, such as CTRL/Z, in a batch job, use the circumflex followed by the appropriate letter, such as ^Z.
$	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.
		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.
		Standard MCR prompt. It may be different for your system.
<	Left angle bracket	General term.

Table 3–1 (Cont.): Keyboard and Keypad Characters

Character	Name	Comments
	Less-than	ASCII name of character. Used in mathematical and logical expressions.
	Period	In text.
	Decimal point	In numbers.
	Dot	In file specifications, delimiter 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 character	Stands for "match one character in this position" in file names and file types.
*	Asterisk	General term.
	Wildcard character	Stands for "match any or no characters in this position" in file names and file types, or for any group, member, or version number.
_	Underscore	Used to tie command words together in DCL.
-	Hyphen	DCL continuation character. Used for command lines too long for terminal line. Overrides RETURN. Command not entered until monitor receives RETURN without preceding hyphen.
	Minus	Mathematical operator for subtraction.
+	Plus	Mathematical operator for addition.
:	Colon	Marks beginning of argument in DCL command.
=	Equal sign	Can also be used preceding argument in DCL command. Also used as logical and mathematical operator in expressions.

Table 3–1 (Cont.): Keyboard and Keypad Characters

Character	Name	Comments
{ }	Braces	General term.
( )	Parentheses	General term. Also, delimiters for a series of arguments or parameters attached to DCL commands or qualifiers
[ ]	Square brackets	Delimiters for UICs and directory names.
/	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 the keys on a calculator. These keys are referred to as 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.

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 RETURN key on the main keyboard.
PF KEYS	Reserved for special functions. They are not used at monitor level.
ARROW KEYS	Used for cursor control by EDT, but are otherwise available for special functions.

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

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
gray blank	types R
PF1	types OP
PF2	types OQ
PF3	types OR
PF4	types OS
FIND	types [1~
INSERT	types [2~
REMOVE	types [3~
SELECT	types [4~
PREV	types [5~
NEXT	types [6~

3.4 The CTRL Key and Control Characters

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

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

When the combination is echoed on your terminal, the CTRL key is represented by the circumflex (^). 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.

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 may have either of two functions on an 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 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.

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. These characters are particularly useful on video terminals where the output may scroll too quickly 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 you use these control characters, none of the output is eliminated from the display. All output that should appear does appear. It is simply held back until you are ready to look at it. (To skip over 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 as /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.

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 Relocatable 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:

- A system-generation option automatically sends a CTRL/U to any terminal that has been awaiting input past a timeout limit. This limit can be set by your system manager in seconds to anything from 1 second to 21 minutes. If this option is included, any line that is not terminated within the timeout limit is wiped out, as indicated by the ^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 timeout limit passes.

- 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.
- 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 function 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:

```
$ THIMK\MK\NK [CTRL/R]
$ 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 type-ahead buffer. This buffer is used to store input characters temporarily until the system can process them.

Occasionally, this buffer gets full. When your type-ahead buffer is full, you cannot type on your terminal and any key you press rings the bell or the keyboard lock light goes 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 type-ahead 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.

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.

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, which may be 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 3.5.1.1 for information on writing help files.

parameter1

If you include a parameter in the HELP command, you will not see the initial HELP display, only the 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 parameters you enter determine the display shown; you do not see intervening displays. See Section 3.5.1.1.

HELP

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 using DCL and do not enter any 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 argument can range from one to 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.

/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 file type, only the file name. You must be logged in to use this qualifier.

Examples

```
$ HELP RET

Help is available for the following DCL commands:

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

 For information on a command, type: HELP commandname.  Additional
 help on a command qualifier is often available by typing:
 HELP commandname qualifier.  For the short forms of some commands,
 type: HELP BRIEF.
```

This example shows the initial display from HELP. The user types HELP without parameters or qualifiers.

```
$ 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 is used to show information. Information on the
 following topics can be obtained 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 The HELP Facility

The HELP facility provides information that most users need in interacting with the system on a daily basis. The online help files make it possible for you to access this information without leaving your terminal.

To use the HELP facility, you type the HELP command. The following information appears on your terminal:

```
For HELP on logging in, type HELP LOGIN.

Help is available for DCL commands, utilities, and system features.
You can receive help by typing either HELP or ? after the DCL prompt,
$, followed by the desired topic.  For example, for help on the COPY
command, type either HELP COPY or ? COPY. (The examples in the HELP
files indicate only the first of these two formats in order to save
space.)  Also note that you can type help requests in either upper or
lower case.

Help is available for the following DCL commands:

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

For information on a command, type: HELP commandname.  Additional
help on a command qualifier is often available by typing:
HELP commandname qualifier.  For the short forms of some commands,
type: HELP BRIEF.

For information on utilities and system features, type: HELP MORE.
```

If you want to see a description of the SET TERMINAL command, you type HELP SET TERMINAL, and the following information appears:

```
SET TERMINAL[:ttnn:]/qualifier[s]

The SET TERMINAL command sets various attributes of your terminal. A
privileged user can set attributes of any terminal.

You can get help for these categories of SET TERMINAL:
```

```
HELP SET TERMINAL COMMON_USE
HELP SET TERMINAL TERMINAL_SETUP
HELP SET TERMINAL TASK_SETUP

Note that the SET TERMINAL/INQUIRE command will set most of your
terminal attributes for you.  See HELP SET TERMINAL INQUIRE.
```

3.5.1.1 Writing Help Files

Complete help files for the RSX-11M-PLUS operating system are provided in the LB:[1,2] system directory, but you can create additional help files for your own use. To do so, you will need to know the following facts about help files and the HELP facility.

A help file is a specially formatted text file. The format is much the same as a conventional outline. The file begins with a general topic followed by subtopics, which in turn are followed by sub-subtopics, up to a total of nine levels. The levels are numbered to enable the HELP facility to find the information quickly. The numbering system is simple and based on the way the user enters the HELP command line at the terminal.

If the user types HELP with no parameters, the HELP facility opens the first help file, called the root file, which is shown in the following example. The introductory section has no number. All other sections have numbers. If the user types HELP and a command name, such as UNLOCK, the HELP facility opens the root file, searches for the number 1 followed by a space and the word UNLOCK, and displays the information on the terminal screen. The following example shows the beginning of the root file for the RSX-11M-PLUS help files and the numbered information on the UNLOCK command. The intervening information has been omitted for clarity.

```
For HELP on logging in, type HELP LOGIN.

 Help is available for DCL commands, utilities, and system features.
 You can receive help by typing either HELP or ? after the DCL prompt,
 $, followed by the desired topic.  For example, for help on the COPY
 command, type either HELP COPY or ? COPY. (The examples in the HELP
 files indicate only the first of these two formats in order to save
 space.)  Also note that you can type help requests in either upper or
 lower case.

 Help is available for the following DCL commands:

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

 For information on a command, type: HELP commandname.  Additional
 help on a command qualifier is often available by typing:
 HELP commandname qualifier.  For the short forms of some commands,
 type: HELP BRIEF.

 For information on utilities and system features, type: HELP MORE.
```

```
1 ABORT

.
.
.

1 UNLOCK

 UNLOCK[/qualifier[s]] filespec[,s]
        /DATE:dd-mmm-yy
        /EXCLUDE:filespec
        /SINCE:dd-mmm-yy
        /SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
        /THROUGH:dd-mmm-yy
        /TODAY

 The UNLOCK command unlocks files that have been locked. A locked
 file is one that was not closed properly (because of a system crash or
 task abort).  Locked files may have been corrupted.

 For additional HELP type HELP UNLOCK qualifier
```

Therefore, any information that you want to have displayed in response to a HELP [parameter1] command is considered level-1 information. Level-1 information must be numbered in the help file as 1, followed by a space and the name of the parameter in uppercase letters with no intervening spaces.

Similarly, if the user types HELP UNLOCK/DATE, the HELP facility opens the root file, searches for the number 1 followed by a space and the parameter UNLOCK and the number 2 followed by a space and the parameter DATE. The slash is ignored by the HELP facility and need not be included in the parameter name in the help file. The following example shows the help file information for the UNLOCK command and the /DATE qualifier.

```
1 UNLOCK

 UNLOCK[/qualifier[s]] filespec[,s]
        /DATE:dd-mmm-yy
        /EXCLUDE:filespec
        /SINCE:dd-mmm-yy
        /SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
        /THROUGH:dd-mmm-yy
        /TODAY

 The UNLOCK command unlocks files that have been locked. A locked
 file is one that was not closed properly (because of a system crash or
 task abort).  Locked files may have been corrupted.

 For additional HELP type HELP UNLOCK qualifier

2 DATE

 UNLOCK/DATE:dd-mmm-yy filespec[,s]

 The /DATE qualifier specifies that you want the UNLOCK command to affect
 only files created on the given date.

2 SINCE
    .
    .
    .
```

The HELP facility displays the information between the level number and parameter it is searching for and the next level number.

The HELP facility searches for up to nine parameters in this fashion. The command HELP invokes the HELP facility, which searches for a line beginning with the number 1 and the first parameter and then continues to search for a line beginning with the number 2 and the second parameter, and so on. After beginning the search for the level-2 parameter, if HELP encounters a line beginning with the numeral 1 before encountering the appropriate level-2 parameter, the search stops, and HELP indicates that no information is available.

Since the root file is apt to become very long if a large amount of level-1 information is included, you may want to put the level-1 information in separate files. The HELP facility will continue the search in a separate file if you include an at sign (@) and the name of the file after the numeral 1 and parameter, in place of the actual information. The following example shows the beginning of the root file and the level-1 information for HELP ANALYZE, which directs HELP to continue the search in a file called DCLANALYZ.HLP. The intervening information has been omitted for the sake of clarity.

```
For HELP on logging in, type HELP LOGIN.

Help is available for DCL commands, utilities, and system features.
                                    .
                                    .
                                    .

For information on utilities and system features, type: HELP MORE.

1 ANALYZE
@DCLANALYZ
```

The following is an example of the file DCLANALYZ.HLP, which contains the information on the ANALYZE command. (The default file type for help files is HLP.)

```
The ANALYZE command allows you to collect information about hardware
on your system.  You can determine the number of bad blocks on a disk
volume with the ANALYZE/MEDIA command, create and format information
on system failures with the ANALYZE/CRASH_DUMP command, and get error
logging information on system devices with the ANALYZE/ERROR_LOG
command.

For more information on these commands, type:

       HELP ANALYZE MEDIA
       HELP ANALYZE ERROR_LOG
       HELP ANALYZE CRASH_DUMP

2 MEDIA
ANALYZE/MEDIA [/qualifier[s]] ddnn:
               /ALLOCATE
               /BADBLOCKS
               /[NO]EXERCISE
               /OVERRIDE
               /RETRY
               /SHOW

The ANALYZE/MEDIA command locates bad blocks on a disk volume and
records their locations in the last good block before the last track
on the disk.  This information is then used by the Backup and Restore
Utility and the INITIALIZE command. See HELP BACKUP and HELP
INITIALIZE.
```

Note that the information on the ANALYZE command in the called-in file has no level number. This is because the level number (1) is in the root file, and HELP begins searching for a file when it encounters the at sign and not a numeral. Numerals can be used for subsequent information, as shown in the example.

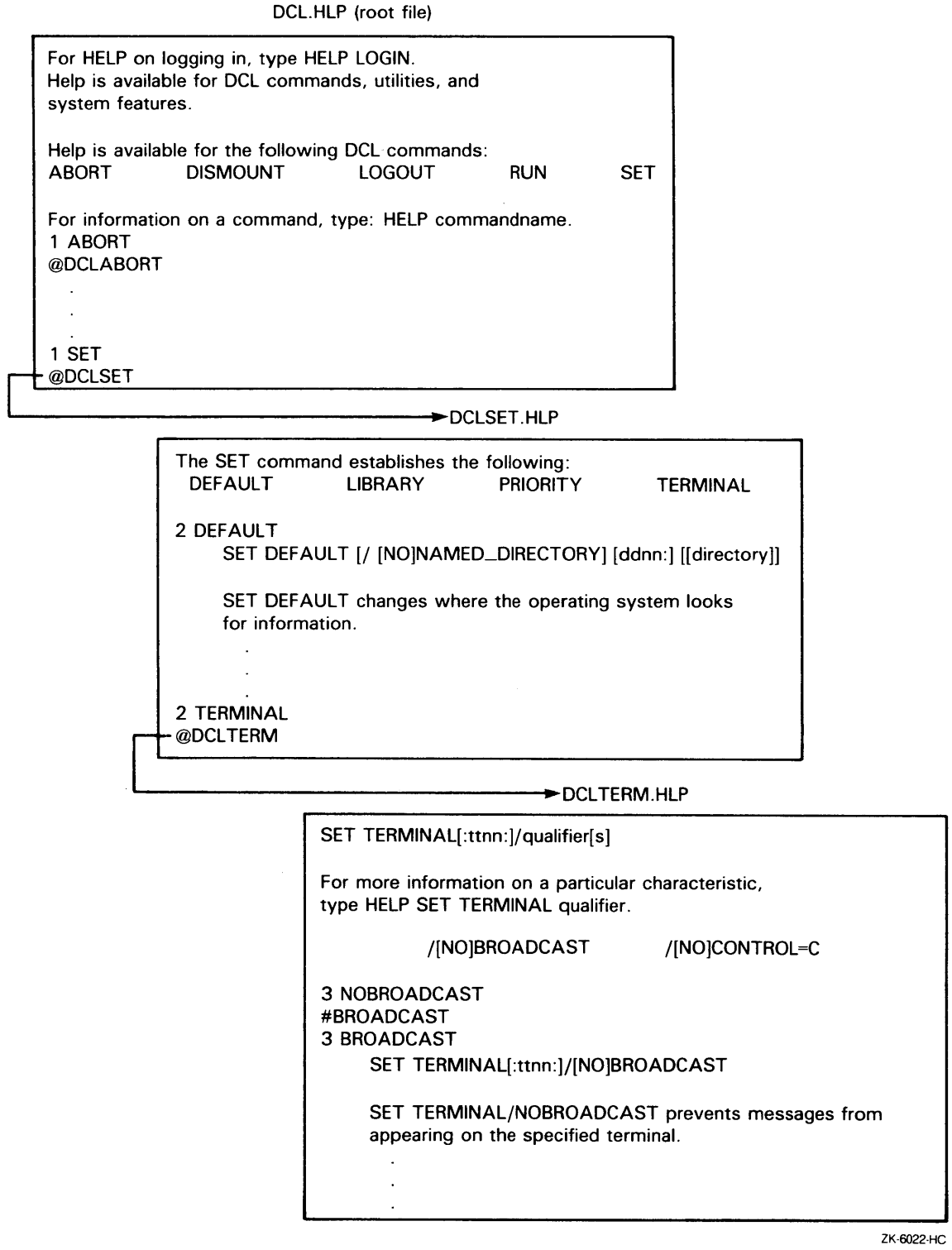
Similarly, if two parameters are synonymous, instead of repeating the information you can enter it only once and direct the HELP facility to display it for both synonyms. To do so, in place of the actual information use the number sign (#) followed by the name of the synonym under which the information is listed, as shown in the following example.

```
1 HELLO
#LOGIN
1 LOGIN

 LOGIN userid

 LOGIN allows you to gain access to the system (if you have a valid
 account) and to begin a user session.

 The userid argument is either your last name or a UIC (user identi-
 fication code). Your UIC consists of two numbers separated by a comma
 (,) or a slash (/). The brackets ([]) that usually surround a UIC
 are optional for this command.

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

Note that the entry for the synonym must precede the entry where the information is actually located.

Figure 3–2 shows the portion of the root file containing help information for the HELP SET TERMINAL command and the files called in by the root file.

Figure 3–2: Help Files for the SET TERMINAL Command



ZK-6022-HC

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 3.7).

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 unless enclosed in quotation marks (").

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).

@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
LOG: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.

/LOG:message

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-87 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]
```

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

```
$ BRO @DAILY [RET]
```

The file DAILY.CMD contains the following messages:

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 EYESHADE. HAVE YOU SEEN IT?

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

BROADCAST

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 -- Message failed to reach TTnn:
```

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

The BROADCAST command uses the breakthrough-write 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—Can't find <username>

Explanation: The specified user is not logged in on any terminal.

User Action: Check the user name for typographical errors. If an error exists, reenter the user name.

BRO—Command input error

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

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

BRO—Command syntax error

Explanation: The command syntax was not correct.

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

BRO—Illegal device specified

Explanation: The destination device was not a terminal.

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

BRO—Message failed to reach TTnn:

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

User Action: Wait and try 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.

BRO—User not receiving messages

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

User Action: Wait and try again.

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-87 13:32   From: TT6:    to: CO:
                    CHARLIE, THE TAPE DRIVE IS DOWN AGAIN

 13:36:22           Login user QUEEQUEG
 14:36:42           Login 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, time-stamped and indented so that it stands out from other messages.

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-PLUS and Micro/RSX System Management Guide* for more information on CO:.

SET TERMINAL

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-PLUS and Micro/RSX I/O Drivers Reference Manual.*

3.8.1 SET TERMINAL

SET TERMINAL sets various attributes of your terminal. Privileged users can set attributes for any terminal. You can determine how all attributes are set for your terminal with the SHOW TERMINAL command, described in Section 3.8.2

Format

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

Parameter

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.

Command Qualifiers

Many of the SET TERMINAL qualifiers have a NO form. 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
/LA12	/LA30P
/LA30S	/LA34
/LA36	/LA38
/LA50	/LA75
/LA100	/LA120
/LA180S	/LA200_SERIES
/LA210	/LN03
/LQP01	/LQP02
/LQP03	/PRO_SERIES
/VT05B	/VT50
/VT52	/VT55
/VT61	/VT100
/VT101	/VT102
/VT105	/VT125
/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	/[NO]TTSYNC
/[NO]TRANSLATION_ROUTINE[:arg]	
	n
	logical

Group 3: Task Setup

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

SET TERMINAL

Command Qualifiers

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

/[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 5 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, VT55, or VT61 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, VT55, or VT61 terminal one line or one screen 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.

Pressing CTRL/C clears hold-screen mode. 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.

/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, LA50, LA200-series, 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.

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_{10} to 255_{10}.

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-PLUS and Micro/RSX 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 to the system a terminal that the /INQUIRE qualifier does not recognize.

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

ASR33	ASR35	DTC01	KSR33
LA12	LA30P	LA30S	LA34
LA36	LA38	LA50	LA75
LA100	LA120	LA180S	LA200_SERIES
LA210	LN03	LQP01	LQP02
LQP03	PRO_SERIES	VT05B	VT50
VT52	VT55	VT61	VT100
VT101	VT102	VT105	VT125
VT131	VT132	VT200_SERIES	

Most of these are DIGITAL terminals. Those with "LA" designations are hardcopy terminals. Those with "VT" designations are video models, as are the PRO-series terminals. Those with "SR" follow the Teletype[1] pattern.

[1] Teletype is a trademark of the Teletype Corporation.

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

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.

/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 0 to 7.

/[NO]DEC_CRT

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

/[NO]EDIT_MODE

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

/[NO]FORMFEED

If the terminal hardware supports form feeds, the terminal should be set /FORMFEED. 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 type-ahead buffer is full. /NOHOSTSYNC, the default, does not lock the keyboard.

The /HOSTSYNC qualifier temporarily locks the keyboard when the type-ahead 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 type-ahead 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 terminals that transmit information in blocks, such as 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 type-ahead buffer. For more information on type-ahead buffers, consult the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.*

/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

Identifies terminals that are not on the automatic set-up list. Your system manager will tell you if you need to use this parameter.

Note to System Managers

The argument n can be any decimal number from 0 to 255 (0— 377_8). The first half of these numbers (0 to 127_{10} or 0 to 177_8) is reserved for use by DIGITAL. Each of the terminals on the automatic set-up list is assigned one of these numbers. The second half (128 to 255_{10} or 177 to 377_8) 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 set-up 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 command.

If you want automatic setup for those terminals not included in the automatic set-up 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.

/[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 /NOTRANSLATION_ROUTINE qualifier disables 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 number of the character- translation routine, 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.

Group 3: Task Setup

The following qualifiers set terminal attributes 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. The process of doing so is not visible to the user. These qualifiers are included for use in cases where the task does not do this setting. See the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual* for information on how tasks can set terminal attributes.

/CHARACTER_LENGTH:n

Sets the length of characters transmitted and received by terminals attached to the system through DH11, DHU11, DHV11, DZ11, or DZV11 variable-speed multiplexers. This qualifier is not valid for terminals attached through DL11 serial-line units. You can set the character length as 7 or 8 bits. The default is 8-bit characters. Nonprivileged users can set the character length for their own terminals only; privileged users can set the character length for any terminal. This qualifier determines the character length that the hardware controller will accept, in contrast to the /[NO]EIGHT_BIT qualifier which determines the character length the software will accept.

/[NO]ECHO

Enables (or disables) echoing of characters typed on the terminal.

The /ECHO qualifier is the default. Characters typed on the terminal are echoed on the terminal.

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

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

/[NO]EIGHT_BIT

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

The /NOEIGHT_BIT qualifier is the default. DIGITAL software employs 7-bit ASCII.

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 8-bit ASCII. Compare with /CHARACTER_LENGTH.

/[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.

/[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). The /NOPASSALL qualifier is the same as /INTERACTIVE. This is the default setting.

/LOCAL
/NOREMOTE

Specifies that the terminal is not attached to a dial-up line. The /NOREMOTE qualifier 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 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 on RSX–11M–PLUS, /NOSERIAL, means that when you enter a command before the previous command has completed execution, the second command is processed in parallel with the first.

When a terminal is set /SERIAL, commands are retained in the terminal's type-ahead 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 type-ahead 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.

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 your terminal is set /NOSERIAL, CTRL/C aborts affect all tasks that have been initiated from your terminal or spawned by tasks initiated from your terminal. (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 qualifier is rarely used. 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 reset the terminal to /NOSLAVE.

/[NO]TYPEAHEAD[:n]

Enables or disables the terminal's type-ahead attribute and also sets the size of the type-ahead buffer. /TYPEAHEAD:66 is the default for instruction- and data-space systems. The type-ahead 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 sounds each time you press a key. You can clear the type-ahead buffer by pressing CTRL/X.

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

On non-instruction- and data-space systems, the size of the type-ahead 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 attributes. 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/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/TYPEAHEAD: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 type-ahead 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 type-ahead buffer.

3.8.2 SHOW TERMINAL

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

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.

The same display results from the following command:

```
$ SHOW TERMINAL/NOSCOPE [RET]
```

The following qualifiers display information about particular types of terminals on the system:

/HT
/RT
/TI
/TT
/VT

The qualifiers /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.

Format

SHOW
Function? TERMINAL[:TTnn:]

SHOW TERMINAL[:TTnn:][/qualifier[s]]

SHOW TERMINAL

Parameter

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, displays all the attributes for your terminal or the terminal you name:

```
$ SHOW TERMINAL:TT6: RET
```

For more examples of the SHOW TERMINAL command, see the end of this section.

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

Command Qualifiers

/ADVANCED_VIDEO
/ANSI_CRT
/AUTOBAUD
/BLOCK_MODE
/BROADCAST
/CHARACTER_LENGTH
/CLI:cliname
/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:n
/PAGE_LENGTH
/[NO]PARITY:arg
ODD
EVEN
/PASSALL
/PASTHRU
/PRINTER_PORT
/PRIVILEGED
/REGIS
/REMOTE
/SCOPE
/SERIAL
/SLAVE
/SOFT_CHARACTERS
/SPEED
/TAB
/TTSYNC
/TYPEAHEAD
/UPPERCASE
/WRAP

Terminal Types

/HT
/RT
/TI
/TT
/VT

Terminal Models

/ASR33
/ASR35
/DTC01
/KSR33
/LA12
/LA30P
/LA30S
/LA34

/LA36
/LA38
/LA50
/LA75
/LA100
/LA120
/LA180S
/LA200_SERIES
/LA210
/LN03
/LQP01
/LQP02
/LQP03
/PRO_SERIES
/VT05B
/VT50
/VT52
/VT55
/VT61
/VT100
/VT101
/VT102
/VT105
/VT125
/VT131
/VT132
/VT200_SERIES

Table 3–2 shows the relationships between SHOW TERMINAL and its displays, and SET TERMINAL. For further information on the displays, see the examples at the end of this section.

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
CHARACTER_LENGTH	CHAR_LENGTH=	CHARACTER_LENGTH:n
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
ESCAPE	ESC	ESCAPE
FORM_FEED	FORM	FORM_FEED

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

SHOW TERMINAL Attribute	SHOW TERMINAL Display	SET TERMINAL Attribute
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
PARITY	PARITY=	PARITY:ODD *or* EVEN
PASSALL	RPA	PASSALL
PASTHRU	PASTHRU	PASTHRU
PRINTER_PORT	PRINTER_PORT	PRINTER_PORT
PRIVILEGE	PRIV=	PRIVILEGE
REGIS	REGIS	REGIS
REMOTE	REMOTE	REMOTE NOLOCAL
SCOPE	CRT	SCOPE NOHARDCOPY
SERIAL	SERIAL	SERIAL
SLAVE	SLAVE=	SLAVE
SOFT_CHARACTERS	SOFT	SOFT
SPEED	SPEED=	SPEED

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

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

Examples

```
$ SHOW [RET]
Function? TERMINAL [RET]
VT4:    [007043]    [7,43]    27-FEB-87 14:10    2       J. KAULLMAN
        CLI    = MCR
        OWNER = none  NOPARITY
        NOLOWER NOCRT
        NOFDX
```

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, and the user's name, plus all the attributes that can be set with SET TERMINAL. See Table 3–2 and the remaining examples.

```
$ SHOW TERMINAL:TT5: [RET]
TT22:   [7,77]   [7,77]    27-FEB-87 11:29      1       J. KAULLMAN
        CLI    = MCR      BUF    = 255.    HFILL = 0       SPEED=(9600:9600)
        LINES = 24.       TERM   = VT100   OWNER = none    NOPARITY
        CHAR_LENGTH = 8 NOPRINTER_PORT  NOPASTHRU        NOSERIAL
        LOWER    NOPRIV  NOHOLD  SLAVE   NOESC   CRT     NOFORM  NOREMOTE
        ECHO     NOVFILL HHT     NOFDX   WRAP    NORPA   NOEBC   TYPEAHEAD
        NOCTRLC AVO      ANSI    DEC     NOEDIT  NOREGIS NOSOFT  NOBLKMOD
        HSYNC    BRO     NOABAUD TTSYNC
```

This example displays all the attributes for another terminal.

```
$ SHO TERM/ESCAPE [RET]
$
```

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

```
$ SHOW TERMINAL/LA36/DCL [RET]
DCL=TT5:
LA36=CO0:
LA36=TT0:
LA36=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 qualifiers 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 it was issued.

```
$ 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 is an acronym for 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 is a mnemonic for ESCape SEQuences.

```
$ SHO TERM/FULL_DUPLEX [RET]
FDX=TT2:
```

```
$ SHO TERM/NOFULL_DUPLEX [RET]
NOFDX=TT1:
    .
    .
    .
```

FDX means full-duplex.

```
$ SHO TERM/PARITY [RET]
PARITY=TT4:ODD
PARITY=TT17:EVEN
```

This example shows all terminals that have parity generation and checking enabled.

```
$ SHO TERM/PASSALL [RET]
RPA=TT1:
    .
    .
    .

$ SHO TERM/INTERACTIVE [RET]
NORPA=TT3:
    .
    .
    .
$ SHO TERM/NOPASSALL [RET]
NORPA=TT3:
    .
    .
    .

$ SHO TERM/NOINTERACTIVE [RET]
RPA=TT1:
    .
    .
    .
```

RPA means Read-Pass-All.

```
$ SHO TERM/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/EIGHT_BIT [RET]
EBC=TT6:
```

EBC means Eight-Bit Characters.

```
$ SHO TERM/CHARACTER_LENGTH [RET]
CHAR_LENGTH=TT10:8
```

The character length shown in the SHOW TERMINAL/CHARACTER_LENGTH display is the character length accepted by the hardware, in contrast to that shown in the SHOW TERMINAL /EIGHT_BIT display, which is the character length accepted by the software.

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 line printer, and alter their contents in various ways. Some error messages issued by 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 in 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 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.

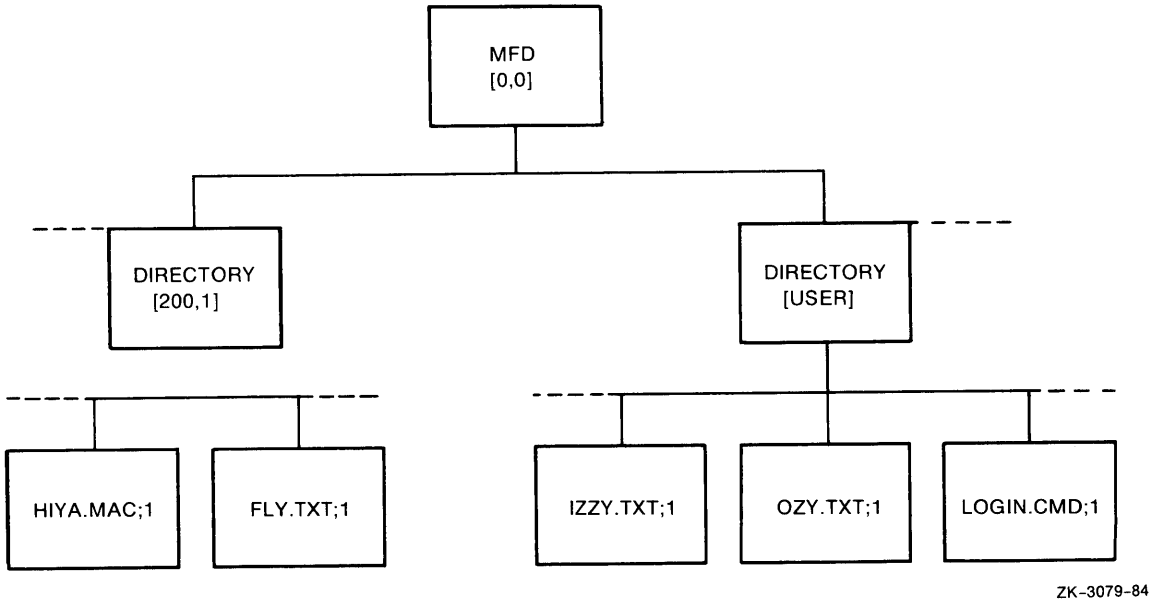
4.1 File Ownership and Location

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 of these commands are described in Sections 5.10 and 12.11.

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–PLUS and Micro/RSX System Management Guide* for more information on ACNT.

Figure 4–1: File Organization on a Volume



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 operating systems are listed in Chapter 5.

[directory]

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

The RSX–11M–PLUS operating system accepts either named directories or numbered directories. The format for a named directory is the following:

[directory] where directory consists of 1 to 9 of the following characters: the letters A to Z and the numbers 0 to 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 to 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, you can specify only numbered directories in the SET DEFAULT command.

name

Specifies the name of the file, which can be from zero to nine of the following characters: the letters A to Z and the numbers 0 to 9.

typ

Specifies the file type. The type consists of one to three of the following characters: the letters A to Z and the numbers 0 to 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.

ver

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

You must separate the file type and version number with a semicolon.

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

If you create a file with version number 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 wildcard characters 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 valid 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 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 TYPE command produce the same desired result:

```
$ TYPE TEXT.TXT
$ TYPE TEXT.TXT;3
$ TYPE TEXT.TXT;0
$ TYPE [MUSTARD]TEXT.TXT;3
$ TYPE DU0:TEXT.TXT
$ TYPE DU0:[MUSTARD]TEXT.TXT
$ TYPE SY:TEXT.TXT
$ 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, because 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.
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 or batch file. System default.
CMF	Compressed 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	FORTRAN–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.
MAP	Task Builder map file. System default.
MLB	Macro library. System default.
OBJ	Object module output from assembler or compiler. System default.

Table 4–1 (Cont.): File Types

File Type	Use
ODL	File containing Overlay Description 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 *RSX–11M–PLUS and Micro/RSX Task Builder Manual* for more information.
POB	Patched object module input for PAT utility; must be specified. See the *RSX–11M–PLUS Utlities Manual* 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 *RSX–11M–PLUS and Micro/RSX Task Builder Manual* 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 Wildcard Characters in File Specifications

In addition to the regular defaults for the current device, the current directory, and the most recent version, you can use wildcard characters 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 [*] RET
```

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 commands DIRECTORY, DELETE, PURGE, COPY, RENAME, TYPE, APPEND, UNLOCK, and SET PROTECTION, a more elaborate form of wildcarding is available. Within file names and file types in these commands, 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.

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 wildcard characters 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.

The following qualifiers can be used to control the time at which a command is executed:

```
/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 the following forms:

```
dd-mmm-yy      as in 25-MAY-87
```

or

```
mm/dd/yy       as in 5/25/87
```

The system always displays dates in the first form.

The same commands also accept the /EXCLUDE:filespec qualifier.

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 character (*). Wildcard characters are accepted for any part of the filespec argument to /EXCLUDE, but you cannot specify just wildcard characters in the file specification.

4.6 Protection

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

The best means of protecting files is to put them on a private disk, diskette, or tape, and keep the device under lock and key. The following section explains how files and volumes can be protected by software on RSX–11M–PLUS systems.

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. However, privileged users may change the protection that you have set and thereby gain access to your files.

4.6.1 File Protection

Each user has a unique User Identification Code—commonly called a UIC—that the system manager assigns when setting up the user's account. Your UIC identifies you to the system.

The following is an example of a UIC:

```
[303,5]
```

In text and command descriptions, the UIC is indicated in either of the following forms:

```
[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 to 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 to 10. Nonprivileged users have group numbers from 11 to 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 *operating system* itself and *privileged users.* A privileged user has a UIC with a group number of 10 or less.

OWNER — The *file owner* 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 *same group number* as the one the file was created under.

WORLD — *All other* users.

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

READ — The user, or the user's tasks, may read, copy, print, or type the file.

WRITE — The user, or the user's tasks, may add new data to the file by writing to it. You cannot edit a file without WRITE access, nor can you use the COPY command to write a file to a directory where you have no WRITE access. If the file is a task image file, WRITE access means you may run it. If you deny WRITE access to a class of users, they cannot run the task.

EXTEND — The user, or the user's tasks, may change the amount of disk space allocated to the file. In practice, EXTEND access is always specified along with WRITE access.

DELETE — The user, or the user's tasks, may delete the file. You cannot delete a file without DELETE access.

Ordinarily, the operating system assigns the 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 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.

The DIRECTORY/FULL command displays the 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.

The SHOW PROTECTION command displays the default protection for all the files that you create.

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 do not 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 is not 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 [RET]
```

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 DUO:[OTHER]
3-JUN-87 15:49

RAMP.TXT;52          (7516,5)      2./2.      23-APR-87 13:12
   [303,5]  [RWED,RWED,RWED,R]

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


Directory DUO:[MOONDOG]
3-JUN-87 15:49

RAMP.TXT;1           (14343,6)     2./2.       3-JUN-87 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-87 15:50

RAMP.TXT;52           (7516,5)     2./2.       23-APR-87 13:12
   [303,5]  [RWED,RWED,RWED,R]

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


Directory DU0:[MOONDOG]
3-JUN-87 15:50

RAMP.TXT;2          (14343,6)      2./2.       3-JUN-87 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 to a directory 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 has access to 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]
```

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, the files are 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–PLUS format on it is called a Files–11 volume. Volume protection works in much the same way as file protection except that it affects the entire volume.

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 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]
$
```

In this example, the user wanted to move a number of files from one directory to another. With CREATE, she 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, CREATE was used to take notes on a phone call.

```
$ CREATE DEVTABLES.TXT [RET]
[CTRL/Z]
```

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

Notes

The function of CREATE differs slightly in batch jobs. See the *RSX-11M-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 (which retypes a line with any corrections you have made), and the DELETE key are the only editing functions available to you when creating sequential files at the terminal.

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
/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 that contains the directory is mounted. This parameter defaults to SY0:.

The volume must be mounted and in Files–11 disk structure.

[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 letters A to Z and the numbers 0 to 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 to 377.

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.

/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
```

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

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

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). A directory can be deleted only after all the files it contains have been deleted.

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 that contains 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.

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.

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 accidently type a file specification that does not exist, EDT prints an error message and returns you to the system command level.

/[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 type 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 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.

EDIT

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 RET
*
```

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 or your previous editing session.

Notes

See the *EDT Editor Manual* for more information on journaling, initialization files, command and key definitions, and the other editing functions of EDT.

The qualifiers used with EDIT/EDT will not work with other forms of the EDIT command. If you use the /EDT qualifier, it must appear immediately after EDIT on the command line. Other qualifiers can float.

You can use the MCR command to invoke EDT in a different format that some users may find more convenient. See the *EDT Editor Manual* for the alternate command format.

EDIT/SLP

4.7.4.2 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-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 to 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 to 14. The default is an audit trail of 8 characters.

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 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 to 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 users may find more convenient. See the *RSX-11M-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–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

Note

Only one qualifier can be specified. This qualifier identifies the editor or function you want to access.

/EDI

Specifies EDI, the Line Text Editor, an interactive, line-oriented editor used to create and maintain text and source files. See the *RSX–11M–PLUS Utilities Manual* for more information on EDI.

/USING:yyy

Specifies that an editor not supplied by DIGITAL is being used, where yyy is the installed name of the editor.

/SOS

Specifies the unsupported editor Son of Stopgap (SOS).

/TECO
/MAKE
/MUNG

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

/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–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.

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:DBO:[7,7]HARRY.MAC:5
*
```

This example invokes SOS for a file that already exists.

```
$ EDIT/SOS LONNNY.MAC RET
Input:DBO:[7,7]LONNNY.MAC;1
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 not unlimited. To conserve space, delete any files that you do not need.

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 wildcard characters 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 character.

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.

Command Qualifiers

Command qualifiers belong to one of the following three groups:

- *Format qualifiers* control the appearance and detail of the directory listing.
- *Destination qualifiers* control where the command output is sent.
- *Other qualifiers* modify the list of files by creation date or exclusion or have other 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, and 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 information:

- File name and type
- Creation date and time
- File organization
- Protection status

- Allocation information
- Last access date and time
- Record format
- Record size
- Primary and alternate key definitions for indexed files
- Bucket size for indexed and relative files

This qualifier invokes the RMS–11 DSP utility. Note that this display is produced for any file you specify in a DIRECTORY command, whether or not the file is an RMS–11 file. Some display information may not be meaningful for non-RMS–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.

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

DIRECTORY

/EXCLUDE:filespec

Specifies that you want the DIRECTORY command not to list certain files. The filespec argument to /EXCLUDE can contain a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

/REWIND

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

Examples

```
$ DIRECTORY RET

Directory DU0:[DOCDAD]
1-DEC-87  14:16

A.A;1                1.          25-SEP-87 12:29
ANIML.BAS;1          6.          24-MAR-87 17:59
AZ.CMD;5             1.          02-APR-87 13:03
BYE.CMD;1            1.          25-SEP-87 12:29
CLEAN.CMD;1          1.          10-JAN-87 08:29
DUN.CMD;4            1.          23-JAN-87 08:05
EDT.CMD;22           1.          07-NOV-87 15:56
LOX.CMD;1            1.          27-APR-87 10:21
LOGIN.CMD;6          1.          06-APR-87 15:56
TI.CMD;4             1.          30-APR-87 08:36
          .
          .
          .
F.TSK;1              4.       C  25-SEP-87 12:30
G.TSK;1              4.       C  25-SEP-87 12:30
PONG.TSK;2           12.      C  25-SEP-87 12:31
BUZZ.TXT;2           4.          12-MAR-87 09:13
IZZY.TXT;1           2.          06-MAR-87 14:33
JIVE.TXT;1           1.          16-NOV-87 13:23

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

This example shows a directory listing in the standard format.

The device and directory are named at the head of the listing, along with the date and the time. All files in the directory are listed because the user did not supply any file specifications. The files shown with a C in column 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-87 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 LB0: [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 LB0:. Notice that the display gives the physical device name, DU0:, for the pseudo device.

```
$ DIR TI.CMD [RET]

Directory DU1:[DOCDAD]
1-MAY-87 14:17

TI.CMD;4            1.          30-APR-87 08:36

Total of 1./5. blocks in 1. file
```

This example displays information on the most recent version of the file named. See the next example.

```
$ DIR TI.CMD;* [RET]

Directory DU1:[DOCDAD]
1-MAY-87 14:17

TI.CMD;4            1.          30-APR-87 08:36
TI.CMD;3            1.          27-APR-87 16:32
TI.CMD;2            1.          27-APR-87 16:29
TI.CMD;1            1.          27-APR-87 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.

DIRECTORY

```
$ DIR/FULL TI.CMD [RET]

Directory DU0:[DOCDAD]
1-MAY-87 14:17

TI.CMD;4              (300,56)       1./5.            30-APR-87 08:36
  [303,5]  [RWED,RWED,RWED,R]    03-MAY-87 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 is that of the file. The first date is the date on which the file was created. The second date is the date of the last revision, followed by the numeral 2, which shows the number of times the file has been revised.

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
```

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-87 13:44
Day of 5-JUN-87

2051CH4.MEM;1      339.      05-JUN-87 10:11
NEWINSDC3.TXT;7    10.       05-JUN-87 10:05
2051CH4.MEM;2      320.      05-JUN-87 10:12
FUT.MAI;1          1.        05-JUN-87 13:05
NEWINSDC3.TXT;6    9.        05-JUN-87 09:55
NEWINSDC3.TXT;10   10.       05-JUN-87 10:39
NEWINSDC4.TXT;3    7.        05-JUN-87 11:46
```

```
FRK.CMD;6          1.          05-JUN-87 13:43
NEWINSDC4.TXT;4    11.         05-JUN-87 13:43
LOG.LOG;1          0.          05-JUN-87 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-87 13:45
*.TXT;* excluded
Day of 5-JUN-87
2051CH4.MEM;1        339.        05-JUN-87 10:11
2051CH4.MEM;2        320.        05-JUN-87 10:12
FUT.MAI;1            1.          05-JUN-87 13:05
FRK.CMD;6            1.          05-JUN-87 13:43
LOG.LOG;1            2.          05-JUn-87 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-1987 12:32      REVISED: 15-JUN-1987 08:19 (5)
FILE PROTECTION:        [RWED,RWED,RWED,R]
RECORD FORMAT:          VARIABLE
RECORD ATTRIBUTES:      CARRIAGE RETURN
FILE ATTRIBUTES:
       ALLOCATION=10    EXTEND QUANTITY=0
```

This example shows the display from the /ATTRIBUTES qualifier to DIRECTORY. Attributes can be displayed for any file, but the full display has meaning only for RMS–11 files.

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

Directory DU1:[LAWRENCEP]
20-MAY-87 09:39

CLOCK.TSK;3          8.     C  15-APR-87 08:55
F.TSK;1              4.     C  25-MAR-87 12:30
G.TSK;1              4.     C  25-MAR-87 12:30
TICTAC.TSK;4         19.    C  01-APR-87 12:07
TREK.TSK;1           54.    C  25-MAR-87 12:31

Total of 89./89. blocks in 5. files
Directory DU1:[ELBODGO]
20-MAY-87 09:39

FRONT.TXT;1          3.          16-MAY-87 11:23
EDITNEWS.TXT;12      25.         11-MAY-87 10:50
OZY.TXT;3            2.          11-MAY-87 10:50
```

```
TEXT.TXT;1          151.        11-MAY-87 10:50
NEW.TXT;1           0.      L   27-APR-87 14:36
AWARE.TXT;2         2.          27-APR-87 13:48
IZZY.TXT;10         2.          12-MAY-87 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-87 09:40

FRONT.TXT;1         3.          16-MAY-87 11:23
EDITNEWS.TXT;12     25.         11-MAY-87 10:50
OZY.TXT;3           2.          11-MAY-87 10:50
TEXT.TXT;1          151.        11-MAY-87 10:50
NEW.TXT;1           0.      L   27-APR-87 14:36
AWARE.TXT;2         2.          27-APR-87 13:48
IZZY.TXT;10         2.          12-MAY-87 13:42
ADVENT.TSK;1        151.    C   13-MAY-87 14:12
DUNGEON.TSK;2       242.    C   11-MAY-87 09:08
PONG.TSK;2          12.     C   13-MAY-87 14:13
YCLOCK.TSK;6        9.      C   08-APR-87 12:17
STAR.TSK;2          102.    C   13-APR-87 09:07
TICTAC.TSK;5        19.     C   01-APR-87 11:31
VTCHS.TSK;12        26.     C   06-MAY-87 15:39
XCLOCK.TSK;6        9.      C   03-MAY-87 16:48
LIFE.TSK.10         4.      C   03-APR-87 13:28

Total of 758./780. blocks in 16. files
```

The command in this example is quite similar to the command used in the previous one, but notice the difference in results. The same user, with default directory [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.

4.8.2 DELETE

DELETE removes 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
/CONFIRM
/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 a wildcard character (*) 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 the following six

characters:

Y	Deletes file named and goes on to next file.
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 character (*). If you do not specify a version number, /QUERY is the default. See the examples.

/CONFIRM

Specifies that you want to decide which files should be deleted on an individual basis. This qualifier is a synonym for /QUERY.

/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 delete certain files. The filespec argument to /EXCLUDE can contain a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/[NO]WARNINGS

Suppresses error messages resulting from the command.

DELETE

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    DUO:[ANNEDIT]WITCH.DOC;2       [Y/N/G/Q]?  Y [RET]
Delete file    DUO:[ANNEDIT]DRY.DOC;4         [Y/N/G/Q]?  N [RET]
Delete file    DUO:[ANNEDIT]PAYCHECK.DOC;1    [Y/N/G/Q]?  Y [RET]
Delete file    DUO:[ANNEDIT]LOADING.DOC;3     [Y/N/G/Q]?  Y [RET]
Delete file    DUO:[ANNEDIT]LOADING.DOC;4     [Y/N/G/Q]?  Y [RET]
Delete file    DUO:[ANNEDIT]LOADING.DOC;5     [Y/N/G/Q]?  Y [RET]
Delete file    DUO:[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]TONTO.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.

Notes

You must have DELETE access to delete a file.

If you want to delete using wildcard characters, 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.

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. Wildcard characters (*) 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.

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 want the PURGE command not to affect certain files. The filespec argument to /EXCLUDE can contain a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/NOWARNINGS
Suppresses error messages resulting from the command.

Examples

```
$ PURGE RET
File(s)? TEMPER.TSK RET
```

In this example, all but the most recent versions of TEMPER.TSK are deleted.

```
$ DIRECTORY ASDIC.TM1:*, ELPASO.TEX;* RET

Directory DU1:[BRUCEW]
20-JUN-87 13:44

ASDIC.TM1;1          1.          20-JUN-87 13:41
ASDIC.TM1;2          1.          20-JUN-87 13:41
ASDIC.TM1;3          1.          20-JUN-87 13:41
ASDIC.TM1;4          0.          20-JUN-87 13:41
ASDIC.TM1;5          1.          20-JUN-87 13:42
ELPASO.TEX;1         1.          20-JUN-87 13:42
ELPASO.TEX;2         1.          20-JUN-87 13:42
ELPASO.TEX;3         1.          20-JUN-87 13:42
ELPASO.TEX;5         1.          20-JUN-87 13:43
ELPASO.TEX;6         1.          20-JUN-87 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 DUO:[GLENN]
20-JUN-87 13:46

ASDIC.TM1;3          1.          20-JUN-87 13:41
ASDIC.TM1;4          0.          20-JUN-87 13:41
ASDIC.TM1;5          1.          20-JUN-87 13:42
ELPASO.TEX;5         1.          20-JUN-87 13:43
ELPASO.TEX;6         1.          20-JUN-87 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.

Note

You can purge any file to which you have DELETE access.

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 that are 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.

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. Wildcard characters 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.

Wildcard characters are acceptable for output files if the destination is another directory. If you have multiple input files and use wildcard characters 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.

/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. The default for this qualifier is to rewind the output tape. If you want to rewind the input tape (or both tapes), you must specify the /REWIND qualifier after the desired devices. For an example, see the end of this section.

/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.

/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 a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

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 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.

```
$ COPY MM1:OLD1.TXT/REWIND MM2:OLD2.TXT [RET]
```

This example rewinds the tape on the input device, MM1:. To rewind the tape on the output device, you would specify the /REWIND qualifier after the output file specification OLD2.TXT.

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.

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 enter a wildcard character (*) for either the file name or the file type, or both. If you use a wildcard character in these fields, you must supply an entry in the version number field. This may be a wildcard character.

If you enter a wildcard character 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.

Wildcard characters leave that portion of the file specification the same as before. No wildcard character 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.

/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 a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/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]
```

In this example, all versions of MAIN.TSK are renamed SUB.TSK. Versions are in the order of creation, with numbers unchanged.

```
$ REN EXHAUST.*;* REFRESH.* [RET]
```

In this example, all files named EXHAUST of whatever type are renamed REFRESH. Their file types remain the same. Note that a wildcard character is given for the input file type. This makes an entry in the version number field mandatory.

RENAME

Notes

You cannot rename files across devices.

Using wildcard characters can be risky. It is wise to experiment before using them on valuable files.

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 wildcard characters 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 wildcard characters are permitted.

CONVERT

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 to 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.

/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 to 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.

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 to 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 number sign, 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 to 377, representing all of the ASCII character set, plus the octal numbers 200 to 377. With the number sign in place, you can use it 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.

RMSCNV reads records from the input file and writes them into successive record cells of the output file, beginning with cell 1. If RMSCNV encounters a cell containing a record, it terminates with an error message. All records written to that point are in the output file. You should examine the two files to determine how far the processing went.

/[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 ouput file is sequential, the output file need not exist before you issue the CONVERT command.

If the output file does not exist, RMSCNV creates an output file with the record attributes of the input file. RMSCNV then reads records from the input file and writes them sequentially into the new output file.

If the output file already exists, and the command does not include the /APPEND qualifier, RMSCNV creates the next higher version of the file. RMSCNV then reads records from the input file and writes them sequentially into the new version of the output file.

If the output file exists and you specify the /REPLACE qualifier, RMSCNV reads records from the input file and writes them sequentially into the output file, starting with the beginning of the file.

If the output file exists and you specify the /APPEND qualifier, RMSCNV reads records from the input file and writes them into the output file, starting with the record position following the last record already in the file.

/[NO]TRUNCATE

Directs RMSCNV to truncate records read from the input file to the output file's record length before writing them into the output file. The trailing bytes of the record are truncated.

The default is /NOTRUNCATE. If you do not specify /TRUNCATE and the input records are too long, RMSCNV terminates with the following message:

```
CON -- Input and output record sizes do not correspond
```

Examples

```
$ CONVERT RET
Input file? FOO.BAR RET
Output file? POOH.BAR RET
```

The output file does not exist. RMSCNV creates the sequential file POOH.BAR and copies the records from FOO.BAR into it.

```
$ CONVERT FOO.BAR POOH.BAR RET
```

This example is the equivalent of the previous one.

```
$ CONVERT/INDEXED GRIZZLY.BAR CROW.BAR RET
```

RMSCNV reads each record of the input file GRIZZLY.BAR sequentially and examines the contents of the primary key field, which is defined as an attribute of CROW.BAR. RMSCNV then inserts the record into CROW.BAR.

```
$ CONVERT/RELATIVE/KEY:1 SAND.BAR MAX.BAR RET
```

RMSCNV reads the indexed input file SAND.BAR, using the first alternate key of the file to establish the sequence of access. Then, RMSCNV writes the records sequentially into an empty relative output file, MAX.BAR, starting with record cell 1.

Note that the /RELATIVE qualifier refers to the organization of the output file and the /KEY qualifier refers to the organization of the input file.

```
$ CONVERT/APPEND/SEQUENTIAL/KEY:2 RUNNING.BAR CORNER.BAR RET
```

RMSCNV reads the indexed input file RUNNING.BAR, using the second alternate key of the file to establish the sequence of access. Then RMSCNV appends the records at the end of the existing sequential output file CORNER.BAR.

If the /APPEND qualifier had not been specified, RMSCNV would have created the next higher version of CORNER.BAR and written the records from RUNNING.BAR into it.

```
$ CONVERT/RELATIVE/FIXED_CONTROL MAMA.BAR PAPA.BAR [RET]
```

RMSCNV reads fixed-length records from the indexed input file MAMA.BAR, using (by default) the primary key of the file to establish the sequence of access. 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. 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 character. If no version is specified, the most recent version is displayed. A wildcard character in any field displays every file that matches otherwise. Both the asterisk (*) and the percent sign (%) wildcard characters 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.

/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 a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/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: SYO:
@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

```
$ 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.
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
$
```

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.

4.9.2 APPEND

APPEND appends to an existing sequential file record 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 wildcard characters 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.

/SINCE:dd-mmm-yy

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

APPEND

/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 a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/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 the 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.

Note

APPEND does not delete the input files. They remain intact. Copies of the input files are placed at the end of the output file.

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 default file type is MAC.

Command Qualifiers

/CHANGE_BAR[:n]

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 n argument is the number of the ASCII character to be used for the change bar. The default is 041, the exclamation point (!). Here are the codes for some commonly used ASCII characters:

! 041

043

$ 044

* 052

+	053
<	074
>	076
?	077
|	174

/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.

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 an 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- to 8-character alphanumeric audit trail symbol. For more information on the Source Language Input Program (SLP), see the *RSX-11M-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 with the exception of one Executive directive being different; version 3 has additional blank lines inserted. In each instance, the "IDENT" number matches the version number.

Here is the text of version 1:

```
        .TITLE WRITE PRINT A PREDEFINED MESSAGE ON THE USER TERMINAL
        .IDENT "01"

        .MCALL ALUN$C, QIOW$C, EXST$C

STRING: .ASCII "This is a simple MACRO-11 program."
LENGTH  =      .-STRING         ; CALCULATE THE STRING LENGTH
        .EVEN

START:  ALUN$C 1,TI,0            ; ASSIGN THE LUN TO USE TO THE TERMINAL
        QIOW$C IO.WVB,1,1,,,,<STRING,LENGTH,40>
                                 ; PRINT THE STRING
        EXST$C EX$SUC            ; SUCCESSFUL EXIT IN ALL CASES

        .END   START             ; SET UP THE STARTING ADDRESS
```

In the following examples, a "difference" as defined by the DIFFERENCES command consists of one or more lines that differ in the two files being compared. Each "difference" ends when the first matching line is encountered. The listing of the "difference" includes all the nonmatching lines AND the first matching line.

```
$ DIFFERENCES/IGNORE:(BLANK_LINES,FORM_FEEDS) [RET]
Filespec1? TOM.MAC;1 [RET]
Filespec2? TOM.MAC;2 [RET]
************************************************
  1  DB2:[303,5]TOM.MAC;1)
    2           .IDENT  "01"
    4           .MCALL  ALUN$C, QIOW$C, EXST$C
    6   STRING: .ASCII  "This is a simple MACRO-11 program."
***************
  2  DB2:[303,5]TOM.MAC;2)
    2           .IDENT  "02"
    4           .MCALL  ALUN$C, QIOW$C, EXST$S
    6   STRING: .ASCII  "This is a simple MACRO-11 program."
          1 differences found
TI:/-BL/-FF/-CB=TOM.MAC;1,TOM.MAC;2
```

In this example, the user compared two files that differ only in one Executive directive—EXST$S or EXST$C—and in their version numbers. The user specified the following qualifier to reproduce the defaults of CMP, the File Compare Utility, which the DIFFERENCES command invokes:

```
/IGNORE:(BLANK_LINES,FORM_FEEDS)
```

The listing of differences between the two files starts with the IDENT numbers on line 2 of the files. The next line is blank in both files. Although the line is numbered, it is not printed on the terminal and is ignored in the comparison. It would not change the results if one file had the blank line there and one did not.

The next listed difference is in the Executive calls on line 4. Again, the blank lines are not shown. The next line of code listed is the same in both files. This marks the end of a group of lines that are different in the two files.

The command line reprinted at the bottom of the listing is the CMP command line equivalent to the DIFFERENCES command.

```
$ DIFFERENCES TOM.MAC;1 TOM.MAC;2 [RET]
************************************************
  1)  DB2:[303,5]TOM.MAC;1
    2           .IDENT  "01"
    3
    4           .MCALL  ALUN$C, QIOW$C, EXST$C
    5
***************
  2)  DB2:[303,5]TOM.MAC;2
    2           .IDENT  "02"
    3
    4           .MCALL  ALUN$C, QIOW$C, EXST$S
    5

    1 differences found
TI:/BL/FF/-CB=TOM.MAC;1,TOM.MAC;2
```

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             ; SET UP 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
**************************************************
  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. For more information on CMP, see the *RSX–11M–PLUS Utilities Manual.*

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 files 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

/EXCLUDE:filespec

Specifies that you want the UNLOCK command not to affect certain files. The filespec argument to /EXCLUDE can contain a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

/NOWARNINGS

Suppresses error messages resulting from the command.

Examples

```
$ Directory TOPEKA.KAN [RET]
Directory DU0:[PBEZ]
5-JUL-87  13:13

TOPEKA.KAN;3    32.     L     25-JUN-87  12:29

Total of 32./40. blocks in 1. files
$
$ UNLOCK TOPEKA.KAN [RET]
$
$ DIRECTORY TOPEKA.KAN [RET]
Directory DU0:[PBEZ]
5-JUL-87  13:14

TOPEKA.KAN;3    32.          25-JUN-87  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

This section describes the PRINT command as well as all the nonprivileged Queue Manager commands.

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 controlled by the Queue Manager (QMG).

Format

PRINT[/qualifier[s]]
File(s) filespec[/qualifier[s]][filespec[,s]]

PRINT[/qualifier[s]] filespec[/qualifier[s]][filespec[,s]]

Command Qualifiers

/[NO]ADJACENT
/[NO]DELETE
/COPIES:n
/DEVICE:ddnn:
/[NO]TRANSFER
/[NO]HOLD
/JOB_COUNT:n
/[NO]JOB_PAGE
/QUEUE:queuename
/[NO]UPPERCASE
/[NO]LOWERCASE
/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

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 character (*) can be used in directory, name, type, and version number fields, but only to stand for the entire field. Wildcarding of the form L*.TXT is not accepted by PRINT, nor is the percent sign wildcard character (%) accepted.

If your file specification includes no file type, the default file type is LST.

Command Qualifiers

/[NO]ADJACENT

Causes print jobs to begin at the point at which the preceding job finished; that is, if the preceding job finished in midpage, the new one will commence on the next line. In print jobs that include multiple files, each file will begin at the point at which the preceding file finished. The /ADJACENT qualifier will override any command qualifiers for file flag pages, regardless of whether or not they are the default. (See the /[NO]FLAG_PAGE and /[NO]JOB_PAGE qualifiers, described in this section.)

The default is /NOADJACENT, which causes a print job to begin on a new page, regardless of where the previous job finished.

Restrictions

1. LN01 and LN03 printers begin print jobs on a new page whether or not the /[NO]ADJACENT qualifier is used, because these printers require a form feed at the end of each print job in order to eject the last sheet in the job.
2. Whether or not the /ADJACENT qualifier is used, LA50, LA75, LA100, LA120, and LA210 printers will begin a print job on a new page if the /FORMS:n qualifier specified for a particular job is different from that specified for the preceding job.

/[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 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

Permits the copying of a file from a private device. The default is /TRANSFER. If a file is transferred from a private device and then printed, the copy of the file is deleted. The original file on the private device is not deleted.

This qualifier can be overridden with a file qualifier.

/DEVICE:ddnn:

Specifies that you want the file printed on a particular device.

/[NO]HOLD

Specifies that you want the 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 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.

/[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:queuename

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 to 32767_{10}. Without this qualifier, the number of pages is unlimited (n=0).

/NAME:jobname

Specifies a name for the print job. Your job name can be up to 9 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.

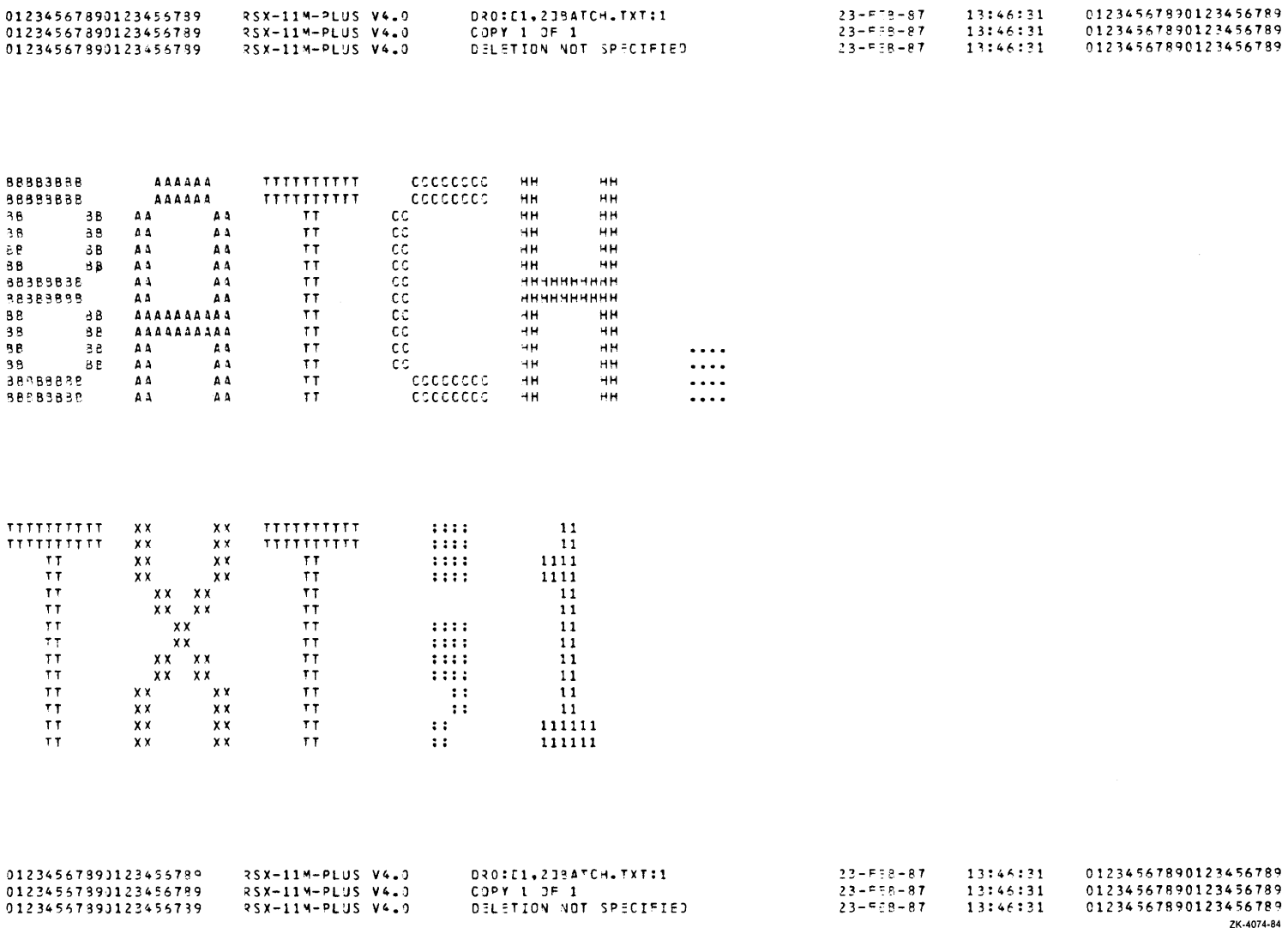
There can be more than one job of the same name, but each job has a unique entry number which appears in the SHOW QUEUE display.

Figure 4–2: Job Flag Page

```
0123456789O123456789    RSX-11M-PLUS V4.0      C7,40JLOGIN     - NO PAGE LIMIT      23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      FORM #0 - NORMAL HARDWARE FORMS      23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      NO IMPLIED FORM FEED                 23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      JOB CONTAINS 2 FILESPECS             23-FEB-87    13:45:49    0123456789O123456789


  SSSSSSSS   MM       MM     IIIIII     TTTTTTTTTT   HH      HH
  SSSSSSSS   MM       MM     IIIIII     TTTTTTTTTT   HH      HH
SS           MMMM   MMMM       II         TT         HH      HH
SS           MMMM   MMMM       II         TT         HH      HH
SS           MM MM MM MM       II         TT         HH      HH
SS           MM  MM  MM       II         TT         HH      HH
  SSSSSS     MM      MM       II         TT         HHHHHHHHHH
  SSSSSS     MM      MM       II         TT         HHHHHHHHHH
        SS   MM      MM       II         TT         HH      HH
        SS   MM      MM       II         TT         HH      HH
        SS   MM      MM       II         TT         HH      HH
        SS   MM      MM       II         TT         HH      HH
SSSSSSSS     MM      MM     IIIIII       TT         HH      HH
SSSSSSSS     MM      MM     IIIIII       TT         HH      HH


LL             OOOOOO       GGGGGGGG    IIIIII     NN      NN
LL             OOOOOO       GGGGGGGG    IIIIII     NN      NN
LL           OO      OO   GG              II       NN      NN
LL           OO      OO   GG              II       NN      NN
LL           OO      OO   GG              II       NNNN    NN
LL           OO      OO   GG              II       NNNN    NN
LL           OO      OO   GG              II       NN  NN  NN
LL           OO      OO   GG              II       NN  NN  NN
LL           OO      OO   GG  GGGGGG      II       NN    NNNN
LL           OO      OO   GG  GGGGGG      II       NN    NNNN
LL           OO      OO   GG      GG      II       NN      NN
LL           OO      OO   GG      GG      II       NN      NN
LLLLLLLLLL     OOOOOO       GGGGGG      IIIIII     NN      NN
LLLLLLLLLL     OOOOOO       GGGGGG      IIIIII     NN      NN


0123456789O123456789    RSX-11M-PLUS V4.0      C7,40JLOGIN     - NO PAGE LIMIT      23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      FORM #0 - NORMAL HARDWARE FORMS      23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      NO IMPLIED FORM FEED                 23-FEB-87    13:45:49    0123456789O123456789
0123456789O123456789    RSX-11M-PLUS V4.0      JOB CONTAINS 2 FILESPECS             23-FEB-87    13:45:49    0123456789O123456789
```

ZK-4075-85

Figure 4–3: File Flag Page



PRINT

/PRIORITY:n

Sets the queue priority of the print job. Nonprivileged users can set the value of n to 0 to 150. Privileged users can set n up to 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.

/FORMS:n

Specifies the type of form on which the job must be printed. The argument n can have a value from 0 to 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 to 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 a job if it is stopped or aborted while being processed.

If you specify /RESTART and the job is interrupted and restarted, the job restarts again from the beginning. With /NORESTART, which is the default, the job restarts at the point at which it stopped.

The job is completely printed in either case, but with /RESTART it is all in one piece.

However, if the paper jams in the line printer, the operator must see that your job is restarted before the point at which the paper jammed.

The /RESTART qualifier and the /DELETE file qualifier cannot be used together because the system deletes the file before the job can be restarted.

If you use the /RESTART qualifier in printing a file that is on a privately mounted disk, you must also use the /NOTRANSFER file qualifier in order to restart the job if it is aborted. The /NOTRANSFER qualifier is required because the system deletes the transferred file before the job can be restarted. In addition, using the /NOTRANSFER qualifier necessitates leaving your private disk attached to the system so that the system can access the job without transferring it.

/[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.

/AFTER:(arg)

dd-mmm-yy hh:mm
mm/dd/yy hh:mm

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 the following 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 omit the calendar field, the day defaults to the current date. The parameter yy must be between 77 and 92. If you omit it, the year defaults to the current year.

The parameter hh:mm is the hour and minute. The Queue Manager always takes a few seconds to start the job. If you leave out the clock field, the time defaults to 00:000:00 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 the next day. You can 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. If you do not specify an exact time, the print job will be blocked until midnight.

File Qualifiers

/COPIES:n

Overrides the /COPIES command qualifier when attached to a particular file specification.

/[NO]DELETE

Overrides the /[NO]DELETE command qualifier when attached to a particular file specification.

/[NO]TRANSFER

Overrides the /[NO]TRANSFER command qualifier when attached to a particular file specification.

PRINT

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/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, PRO.COR [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 PRO.COR.

```
$ PRI/DELETE LASER.TST, PHASER.TST, RAZOR.TST/NODELE [RET]
PRI -- Job 406, name "LASER    ", submitted to queue "PRINT "
```

This example prints the three files and deletes the first two from the current 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-87 13:13) CONST.CON [RET]
PRI -- Job 809, name "CONST    ", submitted to queue "PRINT "
```

This example blocks the job in its queue until the specified date and time. The file itself remains in its directory. Deleting the file from its directory does not take it out of the queue, but does prevent it from being printed.

```
$ PRI/AFTER:(17:00) GUNGA.DIN [RET]
PRI -- Job 765, name "GUNGA     ", submitted to queue "PRINT "
```

This example prints the file after 5 p.m. on the day the command is entered. If the command is entered after the specified time, the file is printed immediately.

Notes

P is the short form of PRINT.

The PRINT command does not accept the percent sign wildcard character (%). The asterisk wildcard character (*) 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.

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-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 in 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 with which restarting or not restarting may be a problem.

If you use the /NOTRANSFER qualifier, you must leave the volume mounted until the print job is finished. Under the default, /TRANSFER, 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 /NOTRANSFER 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 /NOTRANSFER 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 to 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 DELETE

DELETE removes jobs from a queue and files from a job. The command requires either the entry number of the job or the job name and the queue name. Deleting a job by its entry number is preferable to doing so by its name, because job names can be identical but entry numbers are always unique. Also, you need not supply the queue name when you delete a job by its entry number. You can delete a single file in a job by specifying the job name, the queue name, and the position of the file in the job.

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.

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).

```
$ DELETE/JOB LP0 ANGLE RET
```

This command deletes the job ANGLE by first referencing the job's queue, LP0, and then deleting the job ANGLE from LP0.

```
$ DELETE/ENTRY:301/FILE_POSITION:2 RET
```

This example deletes only the second file appearing in job 301. You may refer to the file you want to delete only by the file's position in the job (the numbered order in which you entered the file specification; to determine the position of a file, use the SHOW QUEUE command, described in the next section). You cannot delete a file by its name.

Notes

The DELETE/QUEUE command affects all categories of jobs in any queue. You can delete ACTIVE jobs, WAITING jobs, TIME-BLOCKED jobs, or HELD jobs. For more information on DELETE/QUEUE, see the *RSX-11M-PLUS and Micro/RSX System Management Guide*.

4.10.3 SHOW QUEUE

SHOW QUEUE displays information about print jobs and 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 information for a single queue, either print or batch. The main print queue is called PRINT. The names of other print queues are determined by the system manager.

Command Qualifiers

/FULL

Displays detailed information about queues, queue assignments, jobs, the attributes of jobs in queues, and files in jobs in queues.

/FILES

Displays information about queues, jobs in queues, and files in jobs in queues. The attributes of the jobs are not displayed. This display format is the default for 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.

/ENTRY:nnn

Limits information to a particular job.

/FORMS[:n]

Limits information to jobs that are to be printed on a specified form. If n is omitted, the display shows all jobs on forms other than Form 0.

SHOW QUEUE

/NAME:jobname
Limits information to jobs with the specified job name.

/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, FILES, BRIEF):

```
$ SHOW QUEUE/FULL RET

** PRINT QUEUES **
PRINT  => LP0 (LP1)
  [7,25] ANGLE     ENTRY:22     FORM:0     ACTIVE ON LP0
      PRI:50  LEN:0  PAGE:0  NORESTART  FLAG:JOB  NOLOWER  COP:1
       > 1 DUO:[GREGG]ANGLE.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 DUO:[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 DUO:[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.)

There are two files to be printed in the queue PRINT:

- [7,25]ANGLE.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 jobs 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  => LPO (LP1)
  [7,25] ANGLE     ENTRY:22                        ACTIVE ON LPO
       > 1 DUO:[GREGG]ANGLE.LST;1
  [7,40] ANITA     ENTRY:24
         1 DUO:[LAKE]ANITA.MEM;120
LPO    => LPO
LP1    => (LP1)
** BATCH QUEUES **
BATCH  => BAPO
  [7,40] MJRA      ENTRY:23                        ACTIVE ON BAPO
       > 1 DUO:[7,40]MJRA.BAT;1
```

This display, the default format, shows the queues but does not show the attributes of the jobs. It shows the form number of the job and the attributes of the files if attributes other than the defaults were specified.

```
$ SHOW QUEUE/BRIEF RET

** PRINT QUEUES **
PRINT  => LPO (LP1)
  [7,25] ANGLE     ENTRY:22                        ACTIVE ON LPO
  [7,40] ANITA     ENTRY:24
LPO    => LPO
LP1    => (LP1)
** BATCH QUEUES **
BATCH  => BAPO
   [7,40]MJRA      ENTRY:23                        ACTIVE ON BAPO
```

This display shows only 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.

4.10.4 SHOW PROCESSOR

SHOW PROCESSOR displays information about the initialized characteristics of batch processors, printers, and other output devices controlled by the Queue Manager.

Format

SHOW PROCESSOR [processorname]

Displays information about all processors. This includes batch processors and input devices controlled by the Queue Manager.

SHOW PROCESSOR/BATCH

Shows all batch processors.

SHOW PROCESSOR/PRINT or SHOW PROCESSOR/DEVICE

Shows all print processors.

Examples

```
$ SHOW PROCESSOR RET
```

Information about print processors, batch processors, and other output devices controlled by the Queue Manager (spooled devices) is displayed at your terminal as follows:

```
** 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.

4.10.5 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 for printing 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.

If you supply both the clock and calendar fields, you must separate them with a space.

/FORMS:n

Changes the type of form a job is printed on. See Section 4.10.1 for an explanation of the /FORMS qualifier 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 qualifier as it applies to print jobs.

SET QUEUE

/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 qualifier as it applies to print jobs.

/LOWERCASE

Prints your job on a line printer initialized with the lowercase character set.

/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 qualifier as it applies to print jobs.

/PRIORITY:n

Changes the queue priority of a print job. Nonprivileged users may set priorities up to 150. Privileged users may set priorities up to 250. See Section 4.10.1 for an explanation of the /PRIORITY qualifier to the PRINT command.

/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 a job if it is interrupted while executing. If you specify /RESTART, the job will start again from the beginning. With /NORESTART, the default, the job will restart where it stopped.

/UPPERCASE

Prints the job on a printer initialized with the uppercase character set.

File Qualifiers

/COPIES:n

Changes the number of copies of a file you want to have printed within a a 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 the file position.

Examples

```
$ PRINT/JOBCOUNT:2 JILL.TXT,JOHN.TXT/COPIES:3 [RET]
```

The SHOW QUEUE/FULL command output appears as follows:

```
** 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 appears as follows:

```
** 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 printed instead of three.

Notes

When you issue the PRINT command or the SUBMIT command (for batch processing), you specify attributes of the 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.

4.10.6 STOP/ABORT

STOP/ABORT immediately stops the job currently executing on a line printer or batch processor. Privileged users can stop any job. Nonprivileged users can stop only their own jobs.

Format

STOP/ABORT outputdevice

Parameter

printer

Specifies the processor whose active job you want to stop.

Example

```
$ STOP/ABORT LP0 RET
```

This command stops the currently active job on LP0. This job is deleted from the queue and the next available job is queued.

Note

As soon as the active job is deleted, the Queue Manager passes the eligible job to the processor. The processor has not been aborted, only the job that is active on that processor.

4.10.7 HOLD

HOLD blocks a job in its queue until it is explicitly released.

Format

HOLD/ENTRY:nnn
HOLD/JOB queuename jobname

Notes

Holding a job by its entry number is preferable to doing so by its name, because job names can be identical but entry numbers are always unique. Also, you need not supply the queue name when you hold a job by its 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 must release those jobs with the RELEASE command. For more information on the RELEASE command, see the next section.

4.10.8 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 appears as follows:

```
** 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 DUO:[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 output from the SHOW QUEUE/FULL command appears as follows:

```
** 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 DUO:[AMUSED]MUSE.TXT;1       COP:1        NODELETE
LP0    => LP0
LP1    => LP1
```

Note

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 use SET commands to establish your default device, directory, and User Identification Code, and SHOW commands to display these defaults. You can change the protection code for any existing file in your directory with the SET PROTECTION command. (To display the protection you have set, use the DIRECTORY/FULL command, described in Section 12.8.1.) 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 your default device and directory from those currently set to those indicated in the command.

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 you may have the rights of the GROUP category, depending upon your User Identification Code (UIC). If you are a privileged user, you have the access rights of the SYSTEM category. If your UIC is the directory owner's, 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 a 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 the SET DEFAULT command will accept named directories. Also, the SET DEFAULT command works differently if you previously set your terminal's default to /NONAMED_DIRECTORY, as explained in 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 the name of a physical device, terminate the device name with a colon (:).

If you specify a directory name, you must enclose it in 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, ensure 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 name of the actual physical device, not the logical name.

Command Qualifier

/[NO]NAMED_DIRECTORY

Allows SET DEFAULT to accept either named or numbered directories. The format for a named directory is [directory], and the directory name is one to nine of the following characters: the letters A to Z and the numbers 0 to 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 to 377. Examples of numbered directories are [100,2] and [202,10].

If you issue SET DEFAULT/NONAMED_DIRECTORY, you can specify only 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.

When you specify /NONAMED_DIRECTORY, SET DEFAULT functions as follows:

- You cannot set your default to a named directory. However, you can use named directories in commands.
- You refer to directories in the SET DEFAULT command using the numbered format [g,m].
- The SET DEFAULT command 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.

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.

```
$ 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

```
$ SET DEFAULT [TINA] RET
SET -- Illegal directory

$ SET DEFAULT/NAMED_DIRECTORY [100,100] RET
$ SHOW DEFAULT RET
DUO:[100,100]   Named      TT2:
Protection UIC:    [303,1]

$ SET DEFAULT/NONAMED_DIRECTORY RET
$ SHOW DEFAULT RET
DUO:[303,1]     Nonamed    TT2:
Protection UIC:    [303,1]

$ SET DEFAULT/NAMED_DIRECTORY [DONVAN] RET
$ SHOW DEFAULT RET
DUO:[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.

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 valid, 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 of FCS or RMS–11, supplied with the RSX–11M–PLUS operating system. Note that you need to rebuild your task only 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, 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.

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 your User Identification Code (UIC) and /NAMED_DIRECTORY or /NONAMED_DIRECTORY as your default.

Format

SHOW DEFAULT

Examples

```
$ SHOW DEFAULT RET
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.

4.11.3 SET FILE

SET FILE establishes certain file attributes. You can change an end-of-file marker, change the file identification code (FID) of a file so that it refers to a file in another directory, remove an entry from a directory, or truncate files to their actual length. This command is useful if, for example, you want to use an EDTINI.EDT file in several different directories but do not want to have to change each file when you make additions to the contents. You can create a file with the same name in each directory and give all of them the same FID with the SET FILE command so that the original file is called in each time you begin an editing session.

Format

SET FILE[/qualifier[s]] filespec[,s][/qualifier[s]]

Command Qualifiers	File Qualifier
/ENTER:synonym_filespec	/END_OF_FILE:(BLOCK:n,BYTE:n)
/NOWARNINGS	
/REMOVE	
/TRUNCATE	

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 argument synonym_filespec is the new directory entry. If many users need the same file, this qualifier can save you a lot of disk space. If there is already a file of the name you specify, the new file will be one version number higher than that file.

If you specify more than one file with this qualifier, synonym_filespec refers to all those files. If you omit any field of 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 command.

/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 that contain no files. If you remove the last entry for a file, it can only be found with the VFY utility.

/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 excess 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 marker is. This helps in situations, such as system crashes, in which a file contains useful information but its end-of-file marker is wrong, preventing you from obtaining the information. The end-of-file marker 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 in group or world classification 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 marker past the last byte of the last block allocated to the file.

For BLOCK, the argument n specifies the block number at which the end-of-file marker is to be placed. Usually, the marker 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 marker can be placed in the first byte of the next block. If the number of blocks allocated is 22, 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_8, or 511_{10}.

Examples

```
$ DIRECTORY/FULL HAI.FAI [RET]

DIRECTORY DU1:[HWANG]
16-JUN-87 11:02

HAI.FAI;1      (10,10)      0./55.
16-JUN-87 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 order to check, the end-of-file marker can be

moved out to just past the last block allocated. The user issues the SET FILE/END_OF_FILE command and specifies byte 0 of the first unused block (55 + 1) as the end of the file. This makes it possible to examine the contents of the 55 allocated blocks.

```
$ SET FILE/ENTER: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

Storage used/allocated for Directory DR5:[DIRECT] 4-FEB-87 17:24

$ SET FILE/TRUNCATE *.*;* [RET]
$ DIRECTORY/SUMMARY [RET]

Total of 146./146. blocks in 5. files

Storage used/allocated for Directory DR5:[DIRECT] 4-FEB-87 17:24
```

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 wildcard characters, specifying that all files in the directory are to be truncated. Now there are no unused blocks in the directory. This saves disk space.

4.11.4 SET UIC

SET UIC works differently depending on whether 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 how they can do so. 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 become classified as WORLD, 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]. The first number is the group number and the second is the member number. Group and member numbers are octal and range from 1 to 377.

Group numbers indicate privilege status. The UICs of privileged accounts have group numbers of 10 or less.

Examples

```
$ DIRECTORY/FULL [000000]GIDEON.DIR [RET]
Directory DU0:[000000]
12-JUN-87 14:12

GIDEON.DIR;1        (576,5040)     1./1.     C 12-JUN-87 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 affect 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 a good practice to always use this command.

If you change your UIC to that of a nonprivileged 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.

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 cannot change their UICs, but privileged users can change any 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 to 377.

Generally, 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.

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. Wildcard characters and multiple file specifications that are separated by commas are permitted.

(code)

Specifies which user categories can access a file and how each user category may affect the file. The parentheses are required.

In a file protection code, four categories of users exist, 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 under which the file was created.
GROUP	All users with the same UIC group number as the one under which the file was created.
WORLD	All other users.

SET PROTECTION

Four kinds of access to files exist. The user and the user's tasks may affect the files as follows:

READ	Read, copy, print, type the file, or run it if it is a task image.
WRITE	Add new data to the file by writing to it.
EXTEND	Increase the amount of disk space allocated to the file.
DELETE	Delete the file.

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.

/SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
Specifies that you want to set the protection only on files created within the specified 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 a conventional file specification (for example, TEST.DAT;2) or, if you want to exclude multiple files, a file specification with a wildcard character in all but one of the file specification fields (for example, *.DAT;*, TEST.*;*, or *.*;2). A version number can be given, either explicitly or with a wildcard character (*).

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 DIRECTORY /FULL command displays the protection of files. The SHOW PROTECTION command shows the default protection.

```
$ 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.

If you deny READ access to a task image file, the task cannot be run.

Other commands accepting a protection code include INITIALIZE, INITIALIZE/UPDATE, MOUNT, and CREATE/DIRECTORY.

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-PLUS and Micro/RSX Task Builder Manual*
- MOUNT/OWNER, see Chapter 5
- RUN/UIC, see Chapter 7
- SHOW QUEUE/OWNER_UIC, see Chapter 4

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 how each user category may affect the file. The parentheses are required.

The file protection code has four user categories and four types of access 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.

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 in which 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 set with 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.

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.

4.11.8 SHOW PROTECTION

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 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 [RET]
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 hardware elements that in combination with the CPU and main memory compose a computer system. 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 of 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 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 operating 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 software device driver, which is the interface between the operating system and the device controller.

The table also lists pseudo devices found on RSX–11M–PLUS operating 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.

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)
ADK:	RK05/J/F disk drives
DL:	RL01/RL02 disk drives
DM:	RK06/RK07 disk drives
DR:	RM02/RM03/RM05 disk drives
	RM80 fixed disk drive
DS:	RS03/RS04 fixed disk drives
DT:	DECtape (TU56)
DU:	RA60/RA80/RA81 disk drives
	RA82 disk drive
	RC25 disk drive
	RD31/RD32 disk drives
	RD51/RD52/RD53/RD54 disk drives
	RX33/RX50 diskette drives
DX:	RX01 diskette drive
DY:	RX02 diskette drive
EM:	Nonrotating electronic memory (ML11)
HT:	DECnet host terminal
II:	Interprocessor interface
IP:	IP11 Industrial Control System
LA:	Laboratory peripheral accelerator
LB:	System default pseudo device
LP:	Line printer

Table 5–1 (Cont.): Devices on RSX–11M–PLUS Systems

Mnemonic	Device
LR:	PCL-11 Parallel Communications Link (XCUR)
LT:	PCL-11 Parallel Communications Link (XMTT)
MK:	MKA11 programmable memory box
MM:	TU16/TU45/TU77/TE16 magnetic tape drives
MS:	TS11/TS05/TK25 magnetic tape drives
MT:	TS03/TU10/TE10 magnetic tape drives
MU:	TK50/TK70 magnetic tape drives
	TU80/TU81-E magnetic tape drives
NL:	Null device
PP:	Paper tape punch
PR:	Paper tape reader
RD:	Reconfiguration driver
RT:	DECnet remote terminal
SP:	Spooling pseudo device
SY:	User default pseudo device
TI:	Terminal input pseudo device
TT:	Terminal
VT:	Virtual terminal
WK:	Work-file pseudo device
XM:	DMC-11 interprocessor link

RSX–11M–PLUS operating 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 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.

On RSX–11M–PLUS operating 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 operating systems. Files–11 volumes are those on which the data is organized according to the normal disk structure used by RSX–11M–PLUS operating 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 value. 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 operating systems. These tapes, which are in a nonstandard format, must be mounted using the /NOLABEL qualifier. You can use labeled tapes (the default) for operations that require ANSI format (for example, COPY 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. Backup operations are the most common example of this type of processing. Backup and 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 Files–11 format and 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 and still use high-level languages.

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:)
- DECnet host terminal (HT:)
- DECnet remote terminal (RT:)
- 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:. For more information on pseudo devices, 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 DU1:, both the ALLOCATE and the MOUNT command refer to DU1:. In the ALLOCATE command, DU1: is the device itself; while in the MOUNT command, DU1: is the volume. In most cases, this distinction is not important, but it is helpful to know 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, Logical Unit Numbers, and Logical Names

The purpose of devices of any kind is to handle I/O from tasks. RSX–11M–PLUS operating 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.

You can give any physical device a logical name. 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.

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 DU1:, 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. 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 operating 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–PLUS and Micro/RSX 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:.

Table 5–2: Pseudo Devices on RSX–11M–PLUS Operating Systems

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 Redirection	User default device
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: and have the output sent there rather than to an actual storage device. 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 the end of the 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 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 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.

Pseudo device VT: is used for virtual terminals in batch processing.

The WK: work-file 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 and Micro/RSX 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 to 250 LUNs. The maximum number of LUNs is set when the task is linked.

LUNs can be assigned or reassigned in three ways:

- Any user can specify the ASG option at link time.
- Privileged users can use ASSIGN/TASK to change the LUN assignments in an installed task.
- The task can include the Executive directive ALUN$ (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.

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 the RUN command.

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.)

Logical device assignments can be one of the following three types:

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.

2. Certain login assignments (SY:), 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 out 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 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 a 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:

- 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.)
- 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.
- 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.
- Unowned devices are those that have not been allocated, mounted, or set public. The RSX-11M-PLUS operating system 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 that no longer contains useful information. You must use a combination of several system utilities and DCL commands to prepare a scratch disk for use. This section summarizes the process, which involves the following steps:

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 INITIALIZE/FORMAT command, which invokes the Disk Volume Formatter Utility (FMT). (The *RSX-11M-PLUS Utilities Manual* provides complete information on this utility.) Enter the command as follows, with the appropriate subqualifiers described in Section 5.11.1:

   ```
   $ INITIALIZE/FORMAT ddnn:
   FMT>
   ```

4. Once the disk is ready, you must identify the bad blocks on the disk. These are blocks that are physically unusable. 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.

   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. The following 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.

File	Description
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 has no blocks allocated to it.

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 Ancillary Control Processor (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 5.11.2.)

Certain system functions may also require that the volume be mounted foreign without specifying an ACP. You can check the appropriate documentation to be sure. This requirement may arise if you use the following DCL commands:

- BACKUP
- INITIALIZE

This requirement may also arise if you use the following utilities, described in the *RSX–11M–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 that contains data no longer of use to you. Take the following 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–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 Commands Related to Devices and Volumes

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.

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:

- A volume on a private (allocated) device can be mounted by the owner only.
- 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.
- 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 any of 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. With logical names, 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 the following four logical name tables:

- Task logical name table—Contains logical name entries that are created for an individual task using the Create Logical (CLON$) directive. These entries remain in the table for only as long as the task is running. When either the task has completed execution or the Delete Logical (DLON$) directive has been issued, the logical names are removed from the table. (See the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for more information on the CLON$ and DLON$ directives.)
- 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.
- 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.
- 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
```

This command produces a display of the current logical names in the user logical name table and their equivalence names.

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 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, square 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 must use 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.

You can control the translation of logical names by using the /FINAL qualifier with either the ASSIGN or DEFINE command. For example, you assign three levels of logical names in the following manner:

```
$ ASSIGN DB0:[TSTSYS] TEST5: RET
$ ASSIGN/FINAL TEST5: 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 TEST5:. 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 TEST5: 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 TEST5: (for example, spawning another task, issuing an Assign Channel [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-PLUS and Micro/RSX Executive Reference Manual.*

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. Note that if this command is used on a device that already contains data, the data is erased.

Format

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].

/[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 noted in the bad block descriptor file 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 RETURN instead of CTRL/Z when you finish entering bad block information, the command displays a list of the bad blocks you entered.

```
$ ANALYZE/MEDIA/BADBLOCKS/SHOW DU1: RET
BAD>LBN(S)= 3 RET
BAD>LBN(S)= 347 RET
BAD>LBN(S)= CTRL/Z
BAD -- DU1: 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—Invalid 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 invoked by ANALYZE /MEDIA:

BAD—ddnn: Allocation Failure

Explanation: The ANALYZE/MEDIA command failed to allocate the block number sequence you entered. The I/O failed for a reason other than that the block number was already allocated to another file. This message applies to the /ALLOCATE qualifier only.

User Action: Either the volume is bad or the drive requires maintenance. Try another volume in the same drive.

BAD—ddnn: Bad block file not found

Explanation: The bad block descriptor file could not be read when you used the /BADBLOCKS/NOEXERCISE qualifier.

User Action: You must use the device without updating the bad block file or reformat the device and destroy all data.

BAD—ddnn: Bad block file overflow

Explanation: BAD detected more than 126_{10} 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_{10} characters.

User Action: Reenter the command line.

BAD—Duplicate block number—numb

Explanation: The block number sequence you entered is already present in the bad block file. The value numb is the sequence you entered. BAD ignores any block number sequences you may have entered after the duplicate block numbers.

This message applies only to the /BADBLOCKS qualifier. If this message appears, it means that the block number that was allocated to [0,0]BADBLK.SYS already existed in the bad block descriptor file.

User Action: Reenter the command line with another value.

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 applies to the /ALLOCATE qualifier only.

User Action: The disk must be initialized using the DCL INITIALIZE command.

BAD—ddnn: Failed to read Manufacturer's Detected 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 qualifier.

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 qualifier 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.

BAD—ddnn: Fatal hardware error

Explanation: Your hardware is inoperable.

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 qualifier you have entered is invalid.

User Action: Reenter the command line with a proper qualifier.

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.

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 informational 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–PLUS and Micro/RSX I/O Operations Reference Manual* for an explanation of the error code number.

BAD—ddnn: Volume label incorrect

Explanation: The volume label entered with the /ALLOCATE qualifier 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.

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:.

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 SY0: to the user's system device (assuming it is different from the system device SY:), which is the device that contains the user's files.

/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 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 local, login, 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 the SHOW ASSIGNMENTS command.

You can assign SY0: with SET DEFAULT, described in Chapter 4.

ASSIGN/REDIRECT is a separate command, described in Section 5.4.3.

ASSIGN/TASK is a separate command, described in Chapter 7.

ASSIGN/QUEUE is a separate command, described in the *RSX-11M-PLUS and Micro/RSX System Management Guide.*

Error Messages

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—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?  equivalencename
Logical name?  logicalname

ASSIGN[/qualifier[s]] equivalencename logicalname
```

Command Qualifiers

/FINAL

/GLOBAL
/GROUP[:g]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:
/TRANSLATION_ATTRIBUTES=TERMINAL

Parameters

equivalencename

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.)

logicalname

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 letters A to Z, the numbers 0 to 9, and the underscore (_), colon (:), and dollar sign ($) characters.

The length of the logical name is limited by the maximum 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.

If you enter a device name in the ddnn: format for the equivalence name and logical name, the ASSIGN command will translate the equivalence name to a physical device name and create the logical name with the /FINAL qualifier. The final equivalence name will be the system device or pseudo device name to which the original equivalence name iteratively translated. The logical name and equivalence name must be device names in the ddnn: format for this behavior to occur. If you do not specify a physical device format in both parameters, the equivalence name is not iteratively translated and only undergoes zero compression and colon stripping. The latter behavior allows backward compatibility with logical name behavior in systems without extended logical name support.

ASSIGN

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 (octal) 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 the 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.

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 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 compatibility with the VMS operating system.

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 LB:  RR2: RET
```

This example assigns the logical name RR2: to the pseudo device LB:. 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 DU0:[1,1]SYSLIB.OLB;3. This logical name exists for users whose User Identification Code (UIC) group number is 303.

```
$ 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 command 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 (ddnn:, 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] TEST5 RET
$ ASSIGN/FINAL TEST5 LAST:RET
$ ASSIGN LAST FIRST RET
```

This example shows three assignments. First, the user assigns logical name TEST5 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 TEST5 and then stops. For more information on the iterative translations of logical names, see Section 5.2.3.1.

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 Section 5.4.3.

ASSIGN/TASK is a separate command, described in Chapter 7.

ASSIGN/QUEUE is a separate command, described in the *RSX-11M-PLUS and Micro/RSX System Management Guide.*

Error Messages

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 ASSIGN 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.

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 vice versa. 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 DUO: DU3: RET
$ SHOW DEVICES TYPE DU: RET
DUO:          DU3:          LOADED
DU1:                        LOADED
DU2:                        LOADED
DU3:                        LOADED

$ ASSIGN/REDIRECT RET
FROM? DUO RET
TO? DUO: RET
$ SHOW DEV DU: RET
DUO:                        LOADED
DU1:                        LOADED
DU2:                        LOADED
DU3:                        LOADED
```

This example shows the process of redirection. Following the first ASSIGN/REDIRECT command, 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

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 vice versa.
- 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: The command attempted to establish a circular list of redirections.

User Action: Check for available devices with the SHOW DEVICES command and set up a valid redirection list. Reenter the command.

ASS—Device not redirectable

Explanation: The command attempted to redirect a device that cannot be redirected.

User Action: Check the command line for proper device names and reenter it.

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. Reenter the command after checking for proper device names.

ASS—Old device attached

Explanation: The oldddnn: specified is attached or spooled.

User Action: Determine why the old device is attached and proceed accordingly.

ASS—Old device mounted

Explanation: The oldddnn: specified contains a mounted volume.

User Action: Wait for the device to become available.

ASS—Old device not known to system

Explanation: An ASSIGN/REDIRECT command attempted to redirect an unknown device.

User Action: Check the devices on the system with the SHOW DEVICES command. Reenter the command after checking for proper device names.

ASS—Pseudo device redirect error

Explanation: The 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: The command attempted to redirect pseudo device TI:.

User Action: None. TI: cannot be redirected.

5.5 DEASSIGN

DEASSIGN cancels logical name assignments made by the ASSIGN and DEFINE commands.

Format

DEASSIGN[/qualifier[s]] logicalname

Command Qualifiers

/ALL
/GLOBAL
/GROUP[:g]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL:ttnn:

Parameter

logicalname

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 line 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 letters A to Z, the numbers 0 to 9, an underscore (_), a colon (:), or a dollar sign ($), you must place quotation marks 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, logicalname 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 (:).

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 (octal) 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.

This qualifier is valid only if your system supports extended logical names.

/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.

Examples

```
$ DEASSIGN TPO: 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.

DEASSIGN

```
$ 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 in.

DISMOUNT does not delete logical name assignments.

DEASSIGN/QUEUE is a separate command. See the *RSX-11M-PLUS and Micro/RSX System Management Guide* for more information.

Error Messages

DEA—Device not terminal

Explanation: The /TERMINAL qualifier named a device that is not a terminal.

User Action: Reenter the 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 DEASSIGN/GROUP command.

User Action: Check the group number that you want and reenter the command.

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, directory specifications, or file specifications. If you intend to use a logical name as part of a Files–11 file specification, 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? logicalname
Equivalent name string? equivalencename
DEFINE[/qualifier[s]] logicalname equivalencename

Command Qualifiers

/FINAL
/GLOBAL (or /SYSTEM)
/GROUP[:g]
/LOGIN
/LOCAL
/TERMINAL:ttnn:
/TRANSLATION_ATTRIBUTES=TERMINAL

Parameters

logicalname

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 letters A to Z, the numbers 0 to 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 removed when the logical assignment occurs.

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 placing 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 valid 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.

DEFINE

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 (octal) 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. These local logical names exist only for your terminal.

/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 out of 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 from having to define those logicals each time you log in. For more information on the SYSLOGIN.CMD file, see the description of the HELLO command 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 compatibility with the VMS operating system.

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.

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.

Error Messages

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 include extended logical name support.

User Action: None.

DEFINE

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.

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:

- If you use the form ddnn:, the specified device is allocated (if available).
- 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 the unit number or colon, the first available device with the name dd is allocated.

Examples

```
$ ALLOCATE DU RET
ALL -- DUO:  Now allocated
```

This example allocates the first available DU drive. The message gives the name of the device allocated. This is an example of generic allocation.

ALLOCATE

```
$ 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 allocates the first available RD51 but ignores 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.

DEALLOCATE counteracts ALLOCATE; however, it does not deassign logical names that have been established by 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. See the *RSX-11M-PLUS and Micro/RSX System Management Guide* for information on private devices in batch jobs.

Error Messages

ALL—Device allocated to other user

Explanation: The command attempted to allocate a 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: The command attempted to allocate a device attached by a running task.

User Action: Wait until the task is finished before allocating the device.

ALL—Pseudo device error

Explanation: The command attempted to allocate a pseudo device. Pseudo devices cannot be allocated.

User Action: Reenter the command after checking for the proper device name.

ALL—Public device

Explanation: The command attempted to allocate a public device.

User Action: None. Public devices cannot be allocated.

ALL—User logged on terminal

Explanation: The command attempted to allocate a terminal logged in by another user.

User Action: None. Only terminals that are not logged in can be allocated.

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 nonoperational.

/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 does not deassign any logical names that have been created with the ALLOCATE command.

Error Messages

DEA—Device allocated to another user

Explanation: The 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: The command attempted to deallocate a mounted device.

User Action: You cannot deallocate a mounted device. Dismount the device.

5.9 MOUNT

MOUNT makes a volume known to the system, placing the volume on line and ready for use. Some qualifiers can be used with any MOUNT command; some are limited to mounting disks and other random-access 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
/[NO]CACHE:(arg[,s])
        CREATE[:[region][:[mainpartition][:[size]]]]
        REGION:name
        [NO]DEFER_WRITES
        [NO]DIRECTORY[:extentsize]
        [NO]LOGICAL[:extentsize]
        [NO]OVERLAY[:extentsize]
        [NO]READ_AHEAD[:extentsize]
        [NO]VIRTUAL[:extentsize]
/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
/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-access 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. 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-access 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 blank spaces and the following nonalphanumeric symbols:

! " % _ ' ( ) * + , - . / : < = > ?

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. For more information, see the examples at the end of this section.

Note that the at sign (@), semicolon (;), and dollar sign ($) are not accepted in volume labels.

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.

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–PLUS and Micro/RSX I/O Operations Manual* for further information on the translation from this coding to ANSI label coding.

/FOREIGN

Specifies that the volume being mounted is not in Files–11 format. Such volumes are called foreign volumes. Note that before a disk or tape has been initialized, it is a foreign volume. On RSX–11M–PLUS, this qualifier is required for mounting foreign volumes. If you mount a volume foreign, no one else can access it. If you specify both the /FOREIGN and /SHAREABLE qualifiers when mounting the volume, 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 Ancillary Control Processor (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 the following forms:

- ddnnF1
- ddAACP
- 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.

If the device is a DECtape, MOUNT searches (in the order listed below) until it finds an ACP with a name in one of the following forms:

- ddnnF1
- ddAACP
- 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 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 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 F11ACP if the device is a disk or DECtape, or a copy of 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 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 non-file-structured device QIOs can access the foreign volume.

/PROTECTION:(code)

Specifies the volume protection for Files–11 disks. This protection overrides the volume protection established when the volume was initialized. The /PROTECTION qualifier combines with the /OWNER qualifier to control access to the volume. See Chapter 4 for more information on volume protection.

/PUBLIC
/SYSTEM

These two qualifiers are synonyms. The /SYSTEM qualifier is included for 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 with the /SHAREABLE qualifier 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 with /SHAREABLE and the device is allocated or set public, the device is automatically deallocated or set nonpublic.

A volume mounted with /NOSHAREABLE is dedicated for your private use. No other user can access the volume. For Files–11 volumes mounted with /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.

The display format for disks and other random-access 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–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.

Command Qualifiers for Disks and Files–11 Devices

/ACCESSED:n

Specifies the approximate number of directories that will be accessed simultaneously during the mount. Valid values for n range from 1 to 127. F11ACP 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.

/[NO]CACHE:(arg[,s])

CREATE[:[region][:[mainpartition][:[size]]]]
REGION:name
[NO]DEFER_WRITES
[NO]DIRECTORY[:extentsize]
[NO]LOGICAL[:extentsize]
[NO]OVERLAY[:extentsize]
[NO]READ_AHEAD[:extentsize]
[NO]VIRTUAL[:extentsize]

Associates data caching of disk requests with the specified disk and allows you to override the disk data caching defaults and set one or more of the options. The values to which the options are set with the SET DEVICE/CACHE command become the default values. Subsequent commands change only the options specified in the new command line. The default is /NOCACHE. Note that if you use only one option in a command line, you can omit the parentheses. For more information on data caching, see the *RSX-11M-PLUS and Micro/RSX System Management Guide*. The options are defined as follows:

CREATE[:[region][:[mainpar][:[size]]]]	Creates a cache region in memory and associates caching for the specified device with that region. You can specify the name of the region to be created, the main partition in which the region is to be created, and the size of the region. You specify the size of the region in decimal number of disk blocks. By default, the /CACHE qualifier uses a region called CACHE in the GEN partition with a size of 100_{10} disk blocks.
REGION=name	Associates caching for the specified device with an already existing cache region. The default region is CACHE.
[NO]DEFER_WRITES	Enables or disables data caching of deferred write requests. The NODEFER_WRITES option is the default.
[NO]DIRECTORY[=extentsize]	Enables or disables caching of directory and other volume-structured I/O; that is, logical I/O from an ACP (for example, F11ACP). The default is DIRECTORY. The default extent size for the DIRECTORY option is 1 block.
[NO]LOGICAL[=extentsize]	Enables or disables caching of logical I/O. The default is LOGICAL. The default extent size for the LOGICAL option is 1 block.
[NO]OVERLAY[=extentsize]	Enables or disables caching of overlay I/O. The default is OVERLAY. The default extent size for the OVERLAY option is 4 blocks.
[NO]READ_AHEAD[=extentsize]	Enables or disables reading of the next extent before reading is explicitly requested. The default is /[NO]READ_AHEAD. The default extent size for the READ_AHEAD option is 5 blocks.
[NO]VIRTUAL[extentsize]	Enables or disables caching of virtual I/O. The default is /VIRTUAL. The default extent size for the VIRTUAL option is 5 blocks.

/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 to 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. Valid values for n are the numbers 1 to 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.

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 to 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. Note that the density specified with this qualifier cannot be altered. It can only be checked using the INITIALIZE/UPDATE/DENSITY command.

/[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 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_{10} bytes) because the data did not fill the block, RMS writes an even-aligned (even-byte boundary) block of 18_{10} bytes.
- In the case of blocks less than 18_{10} 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.

/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-PLUS and Micro/RSX 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/PROCESSOR:MRXACP/FOREIGN/PARAMETERS:"MUMBLFRAZ" [RET]
Device? DRO: [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

1. Do 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.
2. Mounting a device resets the error counts and QIOs to zero for that device.
3. 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.

4. There are four types of device ownership, as follows:

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-PLUS and Micro/RSX 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 Ancillary Control Processor (ACP), the magnetic tape Ancillary Control Processor (MTAACP).

The MOUNT command can mount only a single random-access 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.

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.

```
$ MOUNT MS0: FOO [RET]
```

This command mounts a tape set (consisting of a single physical reel) whose File Set ID is FOO and whose Volume ID is FOO. The reel FOO must be loaded on device MS0: at the time of the mount.

```
$ MOUNT MS0: DRAG/VOLUME_IDENTIFICATION:(FOO,BAR,MORE) [RET]
```

This command mounts a tape set (whose File Set ID is DRAG) that consists of three physical reels with Volume IDs of FOO, BAR, and MORE. The reel with the Volume ID FOO must be loaded on device 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 FOO) does not take place until the first record is written to the tape.

```
$ MOUNT/OVERRIDE:SET_IDENTIFICATION MS0: (FOO,BAR,MORE) [RET]
```

This command mounts a tape set (whose File Set ID is immaterial) that consists of three physical reels with Volume IDs of FOO, BAR, and MORE. The reel FOO must be loaded on device MS0:, but the other two reels are not loaded at the time of the mount.

```
$ MOUNT MS0: FOO/VOLUME_ID:(A312,A387,B231) [RET]
```

This command mounts a tape set (whose File Set ID is FOO) that consists of three physical reels with Volume IDs A312, A387, and B231. The reel A312 must be loaded on device MS0:, but the other two reels are not loaded at the time of the mount.

```
$ MOUNT (MS0:,MS1:,MS2:) (FOO,BAR,MORE) [RET]
```

This command mounts a tape set (whose File Set ID is FOO) that consists of three physical reels with Volume IDs FOO, BAR, and MORE. At the time of the mount, the reel FOO must be loaded on device MS0:, the reel BAR must be loaded on device MS1:, and the reel MORE must be loaded on device MS2:.

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 or fewer characters. If your DCL MOUNT command line is more than 80 characters, you can use the MCR MOU command. 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: Reenter the command after checking the device list (SHOW DEVICES) or see your system manager.

MOU—Device specified twice

Explanation: The command specified the same device twice.

User Action: Reenter the 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 the device list (SHOW DEVICES) and find out if the device is in use. See your system manager.

MOU—Home block I/O error

Explanation: An I/O error was detected while the system was 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: Reenter the 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: Reenter the command after checking the 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: Reenter the command after checking the 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 the qualifiers with the SHOW DEVICES command and confirm with the 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 and that the volume has not been initialized. See your system manager.

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 invalid density value.

User Action: Reenter the 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: Reenter the 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: Reenter the 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: Reenter the command after checking for proper syntax.

MOU—Unsupported file header format

Explanation: The volume home block does not conform to Files-11 format. The volume may be corrupted.

User Action: Check to ensure that the volume has not been mounted with the /FOREIGN qualifier. See your system manager.

MOU—Volume already mounted—ddnn:

Explanation: An attempt was made to mount a volume on a device that already had a mounted volume. In multiuser protection systems, only random-access devices can be mounted by multiple users.

User Action: See your system manager.

MOU—Wrong volume label

Explanation: The volume label on the media and the volume label in the command do not match.

User Action: Reenter the command after checking the volume label. Privileged users can use the /OVERRIDE:IDENTIFICATION qualifier.

5.10 DISMOUNT

DISMOUNT marks the volume mounted on the specified device as 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 Ancillary Control Processor (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 command 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 the VMS operating system.

/SAVE

Specifies that the disk is to remain spinning in the drive and can be accessed for reading or writing 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. These values can also be set with the /DEFAULT qualifier to the MOUNT command. 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. With 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. 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.

```
$ LOGOUT [RET]
DMO -- TT1: Dismounted from DU0: *** Final dismount ***
Have a good morning
22-APR-87 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 command was issued. See the 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 DU1: 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 ACP to your terminal indicate only that the volume is marked for dismount. The actual completion of the dismount is noted on the operator's console.

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.

DISMOUNT

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, enter a SET DEVICE:ddnn:/NOCHECKPOINT command and reenter the DISMOUNT command 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: Reenter the command after using the SHOW DEVICES command to check the mount status of devices.

DMO—Volume not mounted by TI:

Explanation: The command attempted to dismount a volume mounted from another terminal.

User Action: Dismount the volume from the proper terminal, or have a privileged user dismount the volume.

DMO—Wrong volume label

Explanation: The command included an incorrect volume label.

User Action: Reissue the command without specifying a volume label or check the volume label to be sure you are dismounting the right volume.

5.11 INITIALIZE

INITIALIZE produces a volume in Files–11 disk structure.

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 5.11.2.) 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 or LOW
          1600 or HIGH
/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

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 6 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 knowledge of the Files–11 file structure. See the *RSX–11M–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. F11ACP maintains a list of the 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 for n is 3. Valid values for n range from 1 to 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.

(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 or LOW
1600 or HIGH

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 checked on an existing volume with the INITIALIZE/UPDATE command. (See Section 5.11.2.)

The acceptable density values for DIGITAL devices with switchable density available on RSX–11M–PLUS are as follows:

Mnemonic	Device	Densities
MM: (magnetic tape)	TE16	800 (default) or 1600
MM: (magnetic tape)	TU16	800 (default) or 1600
MM: (magnetic tape)	TU45	800 (default) or 1600
MM: (magnetic tape)	TU77	800 (default) or 1600
MU: (magnetic tape)	TU81	6250 (default) or 1600
MU: (magnetic tape)	TU81-E	6250 (default) or 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 for 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	Device	File Headers 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
DS:	RS03/RS04	16
DT:	DECtape	16
DU:	RA60	12308
DU:	RA80	7314
DU:	RA81	51699
DU:	RA82	51699
DU:	RC25/RCF25	1565
DU:	RD31	10236
DU:	RD32	20471

Mnemonic	Device	File Headers Allocated Initially (n)
DU:	RD51	664
DU:	RD52	1859
DU:	RD53	4264
DU:	RD54	51699
DU:	RX33	588
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 qualifier 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):

1-header index file:

```
headers = maximum/2
```

headers The initial number of preallocated file headers

maximum The maximum number of files (specified with /MAXIMUM_FILES)

If headers is less than 100_{10}, INITIALIZE preallocates 16_{10} file headers for your volume.

2-header index file:

```
headers = maximum/2 or 25593(10), whichever is larger
```

3-header index file:

```
headers = maximum/2 or 51699(10), whichever is larger
```

See the description of the /MAXIMUM_FILES qualifier for more information on multiheader index files.

/INDEX:arg

BEGINNING
MIDDLE

END
n

Specifies the location of the index 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, as follows:

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 to Z, the digits 0 to 9, a blank space, and the following special characters:

```
! " % & ' ( * ) + , . / : < = > ? ;
```

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 maximum number of files for a disk size of X, use the following calculation:

```
MAXIMUM FILES = (((X-((X+4095.)/4096.)+9.)*127.)/258.)
```

If you do not specify a maximum value, the default value is calculated by the INITIALIZE command using the following formula:

```
DEFAULT MAXIMUM FILES = (theoretical maximum MXF)/8
```

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 Version 2.1. 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:	RA82	65500	65500
DU:	RC25/RCF25	6040	48326
DU:	RD31	2558	20468
DU:	RD32	5117	40942
DU:	RD51	1328	10624
DU:	RD52	3719	29759
DU:	RD53	8529	65500
DU:	RD54	19143	65500

Mnemonic	Device	Default n	Maximum n
DU:	RX33	147	1176
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 directories 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-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.

/OWNER:[g,m]

Specifies the owner of the volume. Group and member numbers range from 1 to 377_8. 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:

```
20MAR8710453
```

The /PROFESSIONAL qualifier locates the system files (index, bitmap, bad block, core image, and directory) at the beginning of the volume instead of in the middle. It also sets the volume owner as [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.

These values are the same as the following DCL qualifiers:

Display	DCL Equivalent
Label	Label
/Bad	/BAD_BLOCKS
/Cha	none
/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.

INITIALIZE

```
$ ALLOCATE DUO: RET
$ MOUNT/FOREIGN DUO: RET
$ INITIALIZE DUO: HONOLULU RET
$ DISMOUNT DUO: RET
$ MOUNT DUO: HONOLULU RET
```

This example illustrates the process necessary to create a volume in Files–11 disk structure on a new disk. (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 disk structure. 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.

```
$ 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 continue the command on a second line.

Note

All users must allocate the device on which the volume is to be initialized and 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 the command with a 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 ANALYZE/MEDIA command and initialize again.

INI—Block(s) exceed volume limit

Explanation: The specified block or blocks exceeded the physical size of the volume.

User Action: Reenter the command after checking for proper qualifier values.

INI—Boot block write error

Explanation: An error was detected in writing out the volume boot block.

User Action: Reenter the command. If it still does not work, see your system manager.

INI—Checkpoint file header I/O error

Explanation: An error was detected in writing out the checkpoint file header.

User Action: Reenter the command. If it still does not work, see your system manager.

INI—Command I/O error

Explanation: INITIALIZE encountered an I/O error while reading the command line.

User Action: Reenter the 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: Reenter the command after checking for proper syntax.

INI—Device allocated to other user - ddnn:

Explanation: You specified a private device not allocated to your terminal.

User Action: Reenter the command after using SHOW DEVICES to check for available devices.

INI—Device not in system

Explanation: The command specified a device not in the current system.

User Action: Reenter the command after using SHOW DEVICES to check for available devices.

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: The command specified a device for which the driver is not loaded.

User Action: Reenter the command after using SHOW DEVICES to check for proper device name.

INI—Failed to attach device—ddnn:

Explanation: INITIALIZE failed to attach the specified device.

User Action: Wait and try again. Use SHOW DEVICES to check for proper device name. 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 with the ANALYZE/MEDIA command and try again.

INITIALIZE

INI—Magtape label must be specified

Explanation: The command attempted to initialize a magnetic tape volume without specifying the required volume label.

User Action: Reenter the command after checking syntax. Magnetic tape volume labels are limited to 6 characters in length.

INI—Magtape write error—ddnn:

Explanation: While writing to the magnetic tape, the system encountered an I/O error.

User Action: Reenter the command.

INI—Not file structured device

Explanation: The system does not support the device named as a Files–11 device.

User Action: Reenter the command after checking proper syntax.

INI—Public device—ddnn:

Explanation: The 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.

INI—Undefined density selection

Explanation: The command specified an invalid density argument for the device named.

User Action: Reenter the command after checking for proper density argument.

INI—Unit is not at high density

Explanation: The command specified that the diskette in the 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: The 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: The command attempted to initialize a volume mounted Files–11.

User Action: Check to ensure you have the right volume.

INI—Volume mounted foreign with ACP

Explanation: The command attempted to initialize a volume mounted /FOREIGN but with the /PROCESSOR qualifier.

User Action: Remount the volume without using the /PROCESSOR qualifier.

INI—Volume name too long—volumelabel

Explanation: The command included a volume label that was too long.

User Action: Reenter the command but limit the length of the volume label for disks or DECtapes to 12 characters; magnetic tapes to 6 characters.

5.11.1 INITIALIZE/FORMAT

The INITIALIZE/FORMAT command formats and verifies volumes on disk cartridges, disk packs, fixed media disks, and diskettes associated with any RSX–11M–PLUS operating system that includes online formatting support in the Executive. (Check with your system manager to determine whether your system includes this feature.) Note that the RX33 diskette is the only DU-type device that can be formatted. For more information on the Disk Volume Formatting Utility (FMT), which this command invokes in order to do its work, see the *RSX–11M–PLUS Utilities Manual.*

Format

INITIALIZE/FORMAT[/qualifier[s]] ddnn:

Parameters

ddnn:

Specifies the device on which the volume is to be formatted.

Command Qualifiers

/BADBLOCKS
/DENSITY:arg
HIGH or DOUBLE
LOW or SINGLE
/ERROR_LIMIT:n
/MANUAL
/[NO]MESSAGE
/OVERRIDE
/[NO]VERIFY
/WRITE_LAST_TRACK:n

Command Qualifiers

/BADBLOCKS

Spawns the Bad Block Locator Utility (BAD) after the formatting operation completes. The BAD operation determines and records the number and location of any unusable blocks. This information is used by the initializing function. If BAD is not installed on the system, the formatting operation prints a warning message on your terminal and exits.

/DENSITY:arg
HIGH or DOUBLE
LOW or SINGLE

Sets RX02 diskettes and RX33 diskettes to either HIGH (DOUBLE) or LOW (SINGLE) density. The default for RX02 diskettes is LOW. The only valid option for RX33 diskettes is HIGH.

/ERROR_LIMIT:n

Sets an error limit, in decimal, for the volume being formatted. If the error count reaches this limit, the command generates an appropriate message and terminates. The default limit is 256 errors. Any value for n greater than 0 but less than or equal to 256 is valid.

/MANUAL

Permits you to format an individual sector or track of a device in manual mode. The INITIALIZE/FORMAT command reads cylinder, track, and sector numbers as decimal unless they are preceded by a number sign (#), which indicates octal values.

In manual operating mode, the following prompts appear on your terminal:

```
** WARNING - Data will be lost on ddn:  **

Continue [Y or N]?
Entering manual mode
Cylinder =
Track    =
Sector   =

Operation complete
```

Restriction

The /MANUAL qualifier cannot be used with DY-type devices.

/[NO]MESSAGE

Specifies that the formatting operation will receive input from an indirect command file that you have created. When you use this qualifier, no operational messages or warnings appear on your terminal, and you may not intervene until the operation is complete.

/OVERRIDE

Causes FMT to ignore the Manufacturer's Detected Bad Sector File (MDBSF) on DM- and DR-type disk volumes. When FMT writes headers on these disks, it normally sets bad sector flags in the headers marked bad in the MDBSF. When the verification procedure discovers a bad sector flag, it reports that the sector was marked in the MDBSF. The /OVERRIDE qualifier inhibits the reporting operation.

/[NO]VERIFY

Determines whether an FMT operation was successful by reading the headers the INITIALIZE /FORMAT command wrote to determine whether they were written correctly. The /VERIFY qualifier is the default.

/WRITE_LAST_TRACK:n

When used with the /VERIFY qualifier on volumes on DM- and DR-type devices, adds a list of the bad sectors the INITIALIZE/FORMAT command found to those indicated by the manufacturer in the Manufacturer's Detected Bad Sector File (MDBSF). The INITIALIZE /FORMAT command also rewrites the header of each bad sector to flag the sector as bad. The n is a required decimal number that is used as the volume's pack serial number.

Restriction

You must specify the /WRITE:n qualifier when formatting volumes on DL-type devices.

INITIALIZE/FORMAT

Examples

```
$ INITIALIZE/FORMAT DB1:/BADBLOCKS RET
```

Formats a volume on a DB-type device and runs the Bad Block Locator Utility (BAD) after formatting is completed.

```
$ INITIALIZE/FORMAT DY2:/DENSITY:HIGH RET
```

Formats a volume on a DY-type device and sets the density to HIGH.

```
$ INITIALIZE/FORMAT DK1:/ERROR_LIMIT:100 RET
```

Formats a volume on a DK-type device and sets the error limit at 100_{10}.

```
$ INITIALIZE/FORMAT DR1:/MANUAL RET
```

Formats a volume on a DR-type device in manual mode. The /MANUAL qualifier permits you to format an individual sector or track on a device. Cylinder, track, and sector numbers are read as decimal unless they are preceded by a number sign (#), which indicates octal values.

```
$ INITIALIZE/FORMAT DM1:/OVERRIDE RET
```

Formats a volume on a DM-type device, ignoring the Manufacturer's Detected Bad Sector File (MDBSF).

```
$ INITIALIZE/FORMAT DP1:/NOVERIFY RET
```

Formats a volume on a DP-type device without verifying whether the formatting operation was successful. The /VERIFY qualifier, which reads the sector headers written by the formatting operation to determine if they were written correctly, is the default.

```
$ INITIALIZE/FORMAT DL1:/WRITE:7 RET
```

Formats a volume on a DL-type device and adds a list of the bad sectors found by the INITIALIZE/FORMAT command to the Manufacturer's Detected Bad Sector File. Also rewrites the headers of bad sectors to flag the sectors as bad.

Error Messages

This section describes the messages that the INITIALIZE/FORMAT command generates and possible user responses.

Command I/O error

Explanation: A hardware transmission error occurred from the keyboard.

User Action: Reenter the command.

Command too long

Explanation: The command was longer than 80 characters.

User Action: Enter a shorter command.

Device does not support formatting

Explanation: A device was specified that does not allow the use of the INITIALIZE/FORMAT command.

User Action: Determine the correct device and if the INITIALIZE/FORMAT operation is valid reenter the command.

Device driver missing

Explanation: The disk device driver is not loaded.

User Action: Load the driver (if it is loadable) and reenter the command, or use a different device in the command line.

Device not in system

Explanation: The specified device was not identified as part of the system during system generation or the device does not exist on the system hardware configuration.

User Action: Determine the correct command line with the correct device mnemonic and reenter the command line.

Device not ready

Explanation: The disk volume was not at operating speed when the INITIALIZE/FORMAT command attempted to access it.

User Action: Allow the volume to reach operating speed and then reenter the INITIALIZE /FORMAT command.

Device offline

Explanation: The device is not in the system hardware configuration.

User Action: Determine the correct command line with the correct device abbreviation and reenter the command line.

Device write locked

Explanation: The volume is write-locked; any write access is prohibited.

User Action: Write-enable the unit and reenter the INITIALIZE/FORMAT command.

Disk is an alignment cartridge

Explanation: The device is a factory-created disk used to align the heads in a disk drive and should not be used for other purposes.

User Action: Use a disk that is not an alignment cartridge and reenter the INITIALIZE /FORMAT command.

Error limit exceeded

Explanation: The number of errors the INITIALIZE/FORMAT command found on the disk exceeded either the number of errors specified with the /ERROR_LIMIT qualifier or the default error limit, 256_{10} errors.

User Action: Set a higher error limit if the /ERROR_LIMIT qualifier was used.

Error reading data

Explanation: The INITIALIZE/FORMAT command encountered an error when it tried to read data from a disk.

User Action: None required. The INITIALIZE/FORMAT command retries the operation and continues the verification.

Error reading header

Explanation: The INITIALIZE/FORMAT command encountered an error when it tried to read a header during a verification operation.

User Action: None required. The task retries the operation and continues the verification.

Error setting diskette density

Explanation: The INITIALIZE/FORMAT command tried to format a diskette but the operation failed.

User Action: Check the syntax and reenter the command with the density reset.

Error writing data

Explanation: The INITIALIZE/FORMAT command encountered an error when it tried to write sector headers.

User Action: None required. The INITIALIZE/FORMAT command retries the operation and continues the verification.

Error writing header

Explanation: The INITIALIZE/FORMAT command encountered an error when it tried to write a header.

User Action: None required. The INITIALIZE/FORMAT command retries the operation.

Failed to attach device

Explanation: The INITIALIZE/FORMAT command could not attach the device to be formatted.

User Action: Determine whether another task has attached the device. If so, wait until the task exits, or abort the task and reenter the command.

Failed to read Manufacturer's Bad Sector File

Explanation: A disk hardware error occurred while the INITIALIZE/FORMAT command tried to read the Manufacturer's Detected Bad Sector File (MDBSF) on the last track of a device.

User Action: Include the /OVERRIDE qualifier and reenter the command.

Fatal hardware error

Explanation: A fatal error occurred in the system hardware configuration.

User Action: Contact your DIGITAL Field Service representative.

Header compare error

Explanation: The INITIALIZE/FORMAT command found an error when it tried to compare headers with the expected value during a verification error.

User Action: None required. The task retries the operation.

Invalid switch

Explanation: An illegal qualifier or a qualifier not valid for the specified device was used in the command.

User Action: Check the syntax and reenter the command.

Manufacturer's Bad Sector File corrupt

Explanation: The factory-written bad block data file, the Manufacturer's Detected Bad Sector File (MDBSF), on the last track of the disk is in an unusable format.

User Action: Reenter the command with the /OVERRIDE qualifier to prevent the INITIALIZE /FORMAT command from trying to use the corrupt bad block data.

Marked bad in Manufacturer's Bad Sector File

Explanation: Indicates that bad block information is recorded in the Manufacturer's Detected Bad Sector File (MDBSF) on the disk.

User Action: None required. This message is for your information only.

Privilege violation

Explanation: The command attempted an operation on a device that was mounted or allocated to another user.

User Action: Use a device that is not mounted or allocated to another user and reenter the command.

Response out of range

Explanation: Parameters entered for manual formatting of an individual sector or track were out of the range for the volume.

User Action: Consult the *RSX-11M-PLUS Utilities Manual* to determine the correct parameters and reenter the command.

Syntax error

Explanation: INITIALIZE/FORMAT detected a syntax error in the command line.

User Action: Determine the correct command syntax and reenter the command.

Unable to run badblock utility

Explanation: The command specified the /BADBLOCKS qualifier, but the Bad Block Locator Utility (BAD) could not be spawned. Either the operating system does not spawn tasks or BAD is not installed.

User Action: Use ANALYZE/MEDIA or BAD separately (see Section 5.3 for more information).

Unrecoverable error - n

Explanation: An I/O error (n) caused the INITIALIZE/FORMAT command to terminate.

User Action: Reenter the command. If the error occurs again, try the command with a different device specified or see the error codes in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual.*

5.11.2 INITIALIZE/UPDATE

The INITIALIZE command creates a volume home block that includes a number of values used by the Ancillary Control Processor (ACP) when you mount the volume. The INITIALIZE /UPDATE command 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 disk structure. In order to use the INITIALIZE/UPDATE command, the INI task must be installed as ...HOM.

Format

INITIALIZE/UPDATE[/qualifier[s]] ddnn: volumelabel

Command Qualifiers

/ACCESSED:n
/DENSITY:arg
HIGH
LOW
/EXTENSION:n
/FILE_PROTECTION:code
/LABEL:newvolumelabel
/MAXIMUM_FILES:n
/OWNER:[g,m]
/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. Valid values for n range from 1 to 127. F11ACP 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

Checks the /DENSITY value established when the volume was initialized. Does not alter the density value in the volume home block. For more information, see the description of the /DENSITY qualifier under the INITIALIZE command (Section 5.11).

/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 to 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 /SHOW. 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:

```
20MAR8710453
```

The /PROFESSIONAL qualifier locates the system files (index, bitmap, bad block, core image, and directory) at the beginning of the volume instead of in the middle and sets the volume owner as [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 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-access 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-87 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. Valid values for n range from 1 to 127.

Note

It is not wise to use the INITIALIZE/UPDATE command without a thorough understanding of the Files–11 disk volume structure and a clear concept of the use the system makes of the values in the volume home block. For more information, see the *RSX–11M–PLUS and Micro/RSX I/O Operations Reference Manual.*

Error Messages

INITIALIZE/UPDATE shares many error messages with INITIALIZE.

<bell> ···ATTENTION··· <bell>

This volume is operating with a replacement home block.

Please take the action outlined in the documentation.

Explanation: The INITIALIZE/UPDATE command 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 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 with 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.

5.12 BACKUP

The BACKUP command transfers a copy of files from a Files–11 disk to tapes or other disks and restores the copy to the original disk when you want to retrieve the files. Its purpose is to safeguard against the loss or corruption of data. When a volume has been backed up, a copy will be available if any of the files on the original volume are lost or corrupted.

The copy that the BACKUP command creates is called a backup set. A backup set consists of all the files backed up with a single BACKUP command. A backup set may extend over several tapes or disks, or several backup sets may be contained on a single tape or disk. Because a backup set is not in Files–11 format, the files must be restored to a Files–11 disk before they can be accessed directly.

The BACKUP command may also be used to copy files from a Files–11 disk directly to another Files–11 disk. The copy that the BACKUP command creates in this case is directly accessible and need not be restored. Note that the output device must be a single Files–11 disk. This disk-to-disk duplication operation cannot utilize two or more output disks.

To perform backup and restore operations, the BACKUP command invokes the Backup and Restore Utility (BRU). Section 5.12.7 lists error messages for BACKUP.

5.12.1 Backup Operations

A backup operation is the transferring of a copy of files from a Files–11 disk to one or more tapes or disks in the form of a backup set. If you look at the format in Section 5.12.5, you will notice that the BACKUP command requires that an input and output device be specified. The type of device you specify for the input and output devices determines the type of operation BACKUP performs.

To back up files to tape, you specify a single Files–11 disk as the input device and up to eight tape drives as the output devices; for example:

```
$ BACKUP DUO: MUO:,MU1:
```

To back up files to disk, you include the /IMAGE:SAVE qualifier to the BACKUP command and specify a single Files–11 disk as the input device and up to eight disk drives as the output devices; for example:

```
$ BACKUP/IMAGE:SAVE DUO: DU1:,DU2:
```

If the number of volumes required for a backup operation exceeds the number of output devices specified, BACKUP prompts you to place additional volumes on the specified drives one at a time.

5.12.2 Restore Operations

A restore operation is the transferring of a copy of files from a backup set on one or more tapes or disks to a single Files–11 disk.

To restore from tape, you specify up to eight tape drives as the input devices and a single disk as the output device; for example:

```
$ BACKUP MUO:,MU1: DUO:
```

To restore from disk, you specify up to eight disk drives as the input devices, a single disk as the output device, and include the /IMAGE:RESTORE qualifier to the BACKUP command; for example:

```
$ BACKUP/IMAGE:RESTORE DU1:,DU2: DUO:
```

5.12.3 Disk-to-Disk Copy Operations

To perform a disk-to-disk copy operation, you specify a single Files–11 disk as the input disk and a single Files–11 disk as the output device; for example:

```
$ BACKUP DLO: DL1:
```

5.12.4 Mounting Disk and Tape Devices for Backup Operations

BACKUP backs up disks that are mounted as Files–11 volumes or as foreign volumes. The disk being backed up must be initialized as a Files–11 volume, however.

BACKUP does not use the Files–11 file system on input disks. 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 disk structure, you must specify the /MOUNTED qualifier. For disks mounted foreign, no qualifier is necessary.

BACKUP also restores to Files–11 or mounted foreign disks. Specify the /INITIALIZE qualifier to restore to a disk mounted foreign. This qualifier initializes the disk to Files–11 disk structure. To restore to a mounted Files–11 volume, specify the /NOINITIALIZE qualifier to indicate to BACKUP that the disk is mounted and in Files–11 format.

Tapes must be mounted foreign, but no qualifier is necessary.

Information on initializing and mounting volumes is summarized in Table 5–3. The BACKUP command returns an error message if the combination of conditions specified is inappropriate.

Table 5–3: Use of /MOUNTED and /INITIALIZE Qualifiers with the BACKUP Command

Volume	Mount Status	Mandatory Qualifier
Input disk	Mounted foreign	None
	Mounted Files–11	/MOUNTED
Output disk	Mounted foreign	/INITIALIZE
	Mounted Files–11	/NOINITIALIZE
Input tape/ Output tape	Mounted foreign	None

5.12.5 The Backup Command Line

General Format:

BACKUP[/qualifier[s]]
From? indevice:[filespec[,s]]
To? outdevice:

BACKUP[/qualifier[s]] indevice:[filespec[,s]] outdevice:

Parameters

indevice:

Specifies the input device you want to transfer files from. In a backup operation, the input device contains the files you want to safeguard. In a restore operation, the input device contains the backup set you are restoring.

There is no default for the input device. It must be specified.

For a restore operation, up to eight input devices may be specified, separated by commas. BACKUP restores the backup set beginning on the first device and continuing on the other devices in the order specified. If the number of volumes in the backup set you are restoring exceeds the number of input devices specified, BACKUP prompts you to place additional volumes on the specified drives, one at a time.

outdevice:

Specifies the output device you want to transfer the files to. In a backup operation, the output device contains the backup set you want to create. In a restore operation, the output device is the disk that receives the files you are restoring.

There is no default for the output device. It must be specified.

In a backup operation, up to eight output devices may be specified, separated by commas. BACKUP creates the backup set beginning on the first device and continuing on the other devices in the order specified. If the number of volumes required for the backup set exceeds the number of output devices specified, BACKUP prompts you to place additional volumes on the specified drives, one at a time.

[filespecs]

The optional 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;version
```

You may specify as many as 16 file specifications per command, separating the file specifications by a comma. The file specification always follows the input device. If more than one input device is specified, the file specification must follow the first input device. Note that file specifications may not be placed after the output device.

When you enter a command without a file specification, all the files on the input volume are copied to the output volume.

Files can also be backed up or restored selectively by directory, file name, file type, or version number. If only the directory is specified, the remainder of the file specification will be treated as a wildcard; for example, *[directory]* is equivalent to *[directory]*.*;**. If the version number is omitted, a wildcard is the default; for example, *filename.typ* is equivalent to *filename.typ;**. Note, however, that BACKUP does not accept 0 or –1 as version numbers.

See the next section for a list of the qualifiers available with the BACKUP command.

5.12.5.1 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.

The six groups of qualifiers are as follows:

Group 1: Selective Backup and Restore Qualifiers

Command Qualifiers

/CREATED:BEFORE:(dd-mmm-yy nn:mm)
AFTER:(dd-mmm-yy nn:mm)
/EXCLUDE
/MODIFIED:arg
BEFORE:(dd-mmm-yy nn:mm)
AFTER:(dd-mmm-yy nn:mm)

Group 2: Initialization Qualifiers

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

Group 3: Tape Control Qualifiers

Command Qualifiers

/APPEND
/DENSITY:arg
800
1600
/ERROR_LIMIT:n
/LENGTH:n
/REWIND
/LABEL:TAPE:fileset-ID
/SAVE_SET:name

Group 4: Verification Qualifiers

Command Qualifiers

/COMPARE
/VERIFY

Group 5: Display Qualifiers

Command Qualifiers

/IDENTIFICATION
/[NO]LOG
/LIST

Group 6: Disk Processing Qualifiers

Command Qualifiers

/APPEND
/DIRECTORY
/IMAGE:arg
SAVE
RESTORE
/NOINITIALIZE
/LABEL:INPUT:volumelabel
/LABEL:OUTPUT:volumelabel
/MOUNTED
/NEW_VERSION
/[NO]PRESERVE
/[NO]REPLACE
/SAVE_SET:name

5.12.5.2 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-87
/AFTER:14:00
/BEFORE:(1-APR-87 12:00)
/AFTER:(1-APR-87 12:00)
/BEFORE:4/1/87
/AFTER:(4/1/87 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.

/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].

/MODIFIED:arg
BEFORE:(dd-mmm-yy nn:mm)
AFTER:(dd-mmm-yy nn:mm)

Specifies selection of files modified before or after the specified date, 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:

```
/BEFORE:31-MAY-87
/AFTER:14:00
/BEFORE:(1-APR-87 12:00)
/AFTER:(1-APR-87 12:00)
/BEFORE:4/1/87
/AFTER:(4/1/87 12:30)
```

Note that if you want to enter both a date and a time, you must use the parentheses.

5.12.5.3 Initialization Qualifiers

This section describes the qualifiers you may 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. There is no default for this qualifier; you must specify either /INITIALIZE or /NOINITIALIZE. If you do not, you will be asked whether or not you want to initialize before proceeding. Note that the disk must be mounted foreign and that you cannot initialize a device mounted by another user.

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 disks using the DCL command ANALYZE/MEDIA.

AUTOMATIC specifies that bad block information for the volume be determined automatically from the bad block descriptor file, created by the ANALYZE/MEDIA command. This is the default.

MANUAL specifies that bad block information for the volume is to be entered manually.

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.

/ACCESSED:n

Specifies the default number of File Control Blocks for each 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 Section 5.11 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 5.11. 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.
- END—Specifies the end of the volume.
- n—Specifies the logical block n.

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 5.11 for more information.

5.12.5.4 Tape Control Qualifiers

The following qualifiers allow you to control the way the operating system manages tape data transfer operations.

/APPEND

Adds new data to a volume that already contains one or more backup sets. If only one volume is required to contain a single backup set, any space remaining on that volume may be used for additional backup sets. However, if more than one volume is required and space remains on the last volume, this space cannot be used for additional backup sets, because BACKUP cannot restore them. Use of the /APPEND qualifier is therefore restricted to the first volume.

If the tape is not positioned at the beginning, the /APPEND qualifier causes the output to be written at the point at which the tape is currently positioned. If you want to begin 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.

/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 value for n is 25.

/LENGTH:n

Specifies the length of the output tape in decimal feet. You may want to use this qualifier when you know that the last part of a tape is defective and must not be used. You can specify a shorter length and ensure that you do not write on that part of the tape.

The default length is the physical length of the output tape.

Restriction

The /LENGTH qualifier cannot be used with the TK25 and TK50 tape cartridge devices.

/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.

/SAVE_SET:name

Allows you to create a different name for each backup set that you place on tape or disk. The name can be up to 12 characters long.

When you place several backup sets on a tape or disk output volume, each backup set should have a different name in case you need to restore your backup sets. If the backup sets have the same name, only the first backup set created can be restored by BACKUP.

If you do not use this qualifier, the backup set name 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, 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 restored. 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 restore operation, BACKUP searches the first disk for each backup set you specify. Each backup set is then restored in the order you specified.

Note that all backup sets must begin on the first tape or disk in your set. For example, if you have six backup sets on tape or disk, the first five backup sets must begin and end on the first reel of tape, or on the first disk. The sixth backup set must also begin on the first reel of tape, or the disk, but it may span more than one tape or disk.

5.12.5.5 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.

This restriction does not apply to the /[NO]INITIALIZE qualifier.

/VERIFY

Verifies that the output volume was written correctly by comparing the input volume to the output volume and reporting any differences.

During a backup operation, each tape or disk is verified before starting the next volume in the backup set. During a restore operation, however, the entire backup set is restored before beginning a verify operation.

Restriction

You cannot specify the /NEW_VERSION and /VERIFY qualifiers together. BACKUP cannot verify a restore operation when the /NEW_VERSION qualifier is used because the version numbers of the files on the output device may not correspond to the version numbers of the files on the input device.

5.12.5.6 Display Qualifiers

The following qualifiers display information about the data transferred.

/IDENTIFICATION
Directs BRU to identify itself by displaying its software version number.

/[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 names of the backup sets or files on the specified volume. In a multivolume backup set, the directory is on the first tape or disk of the set. See the examples in Section 5.12.6.

5.12.5.7 Disk Processing Qualifiers

The following qualifiers allow you to control the way the operating system manages disk data transfer operations.

/APPEND
Adds new data to a volume that already contains one or more backup sets. If only one volume is required to contain a single backup set, any space remaining on that volume may be used for additional backup sets. However, if more than one volume is required and space remains on the last volume, this space cannot be used for additional backup sets, because BACKUP cannot restore them. Use of the /APPEND qualifier is therefore restricted to the first volume.

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.

You cannot use the /APPEND qualifier during a backup operation to a mounted disk.

/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.

/IMAGE:arg
SAVE
RESTORE
Specifies that you want to use more than one 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. 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. 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 performing a disk-to-disk copy operation (an operation in which the input and output devices are single Files–11 disks). For these operations, use the /INITIALIZE qualifier.

/NOINITIALIZE

Specifies that you do not want to initialize the output disk, because it already contains a Files–11 structure and is mounted as a Files–11 volume.

/LABEL:INPUT:volumelabel

Specifies the volume label of the input disk. This is not a required parameter, but if you specify it, BACKUP checks the volume label.

/LABEL:OUTPUT:volumelabel

Specifies a volume label for the output disk. The default is the label of the input volume.

/MOUNTED

Allows you to back up files from a disk that is mounted as Files–11.

/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.

Restriction

You cannot specify the /NEW_VERSION and /VERIFY qualifiers together. BACKUP cannot verify a restore operation when the /NEW_VERSION qualifier is used because the version numbers of the files on the output device may not correspond to the version numbers of the files on the input device.

/[NO]PRESERVE

Specifies whether you want BACKUP to preserve file identifiers during a transfer operation using disks. The default is /PRESERVE.

/[NO]REPLACE

Specifies whether or not files from the input volume (which contains the backup set) should replace files having identical file specifications on the output volume during a restore operation.

A conflict occurs when a file on the input volume and a file on the output volume have identical file specifications. The default qualifier /NOREPLACE resolves that conflict by keeping the file on the output volume rather than replacing it with the backup copy of the file on the input volume. A message informs you that the file has not been superseded. The /REPLACE qualifier resolves the conflict by deleting the file on the output volume and replacing it with the file being restored from the input volume.

/SAVE_SET:name

Allows you to create a different name for each backup set that you place on tape or disk. The name can be up to 12 characters long.

When you place several backup sets on a tape or disk output volume, each backup set should have a different name in case you need to restore your backup sets. If the backup sets have the same name, only the first backup set created can be restored by BACKUP.

If you do not use this qualifier, the backup set name 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, 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 restored. 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 restore operation, BACKUP searches the first disk for each backup set you specify. Each backup set is then restored in the order you specified.

Note that all backup sets must begin on the first tape or disk in your set. For example, if you have six backup sets on tape or disk, the first five backup sets must begin and end on the first reel of tape, or on the first disk. The sixth backup set must also begin on the first reel of tape, or the disk, but it may span more than one tape or disk.

5.12.6 Examples

The following are examples of various BACKUP operations.

Examples

```
$ BACKUP/LIST MS0: RET

VOL1.  18JUL87  INPUT1  19-JUL-87 15:28:28
VOL1.  BACKUP2  INPUT2  19-JUL-87 15:28:44
VOL1.  19JUL87  RANDOM  19-JUL-87 15:28:58
```

This example shows a directory of a tape volume. It displays the tape volume number (VOL1.), the names of the three backup sets contained on the tape, the names of the three volumes backed up, and the date and time of the backups.

```
$ BACKUP/SAVE_SET:19APR87/LIST DU1: RET
VOL1 19APR87  RANDOM  19-APR-87  23:37:11
[303,13]
27DECE.LST;1
ASTCRSH.MAC;2
APNDXC.TXT;1
X.MAC;1
X.OBJ;1
X.TSK;1
X.MAP;1
[306,6]
27DECE.CDA;1
RSX11M.STB;3
```

```
BAC - Completed on 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:18APR87/NOINITIALIZE MS0: DU0: RET
```

This example rewinds the tape on drive MS0:, then locates the backup set with the name 18APR87 (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/REVISED:AFTER:(14-MAY-87 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, 1987. 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 DU1:
BAC - Completed
```

This example shows a backup of the fixed disk to several disks. You must use the SAVE option with the /IMAGE qualifier when doing a multiple-disk backup operation.

```
$ BACKUP/INI/IMAGE:RES DU1: DUO: 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 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 - 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 disk restore operation to the fixed disk DU0:. You must specify the RESTORE option on the command line with the /IMAGE qualifier when restoring several disks to the fixed disk.

5.12.7 BACKUP Error Messages

This section lists BACKUP error messages, describes the meaning of each message, and suggests actions to correct the errors. Error messages from the BACKUP command are preceded by the mnemonic BAC.

BACKUP has three kinds of error messages: informational, warning, and fatal. They are described as follows:

- Informational messages provide information on the current state of your backup operation.
- Warning messages are marked *WARNING*. They inform you of conditions that require your consideration. Depending on the message and the operation, you decide what further action is called for. The operation continues.
- Fatal messages are marked *FATAL*. They inform you of conditions that caused BACKUP to terminate execution of your command. The operation does not complete.

The subsections that follow describe each type of message. The messages are listed in alphabetical order.

5.12.7.1 Informational Messages

BAC—BRU version xx.xx

Explanation: This message identifies the version of the Backup and Restore Utility (BRU) being used.

User Action: None.

BAC—Completed

Explanation: The BACKUP operation is complete.

User Action: Enter another BACKUP command line or exit by typing CTRL/Z.

BAC—End of disk n on ddnn

Explanation: BACKUP finished transferring data or verifying the disk n on device ddnn.

User Action: None.

BAC—End of tape n on ddnn

Explanation: BACKUP finished transferring data or verifying the tape n on device ddnn.

User Action: None.

BAC—Mount another disk

Explanation: BACKUP is requesting that a new disk be mounted after encountering a fatal disk write error.

User Action: Mount a new disk on the drive.

BAC—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.

BAC—Mount disk n on ddnn:. Press "RETURN" when done

Explanation: BACKUP is requesting a disk for a backup or restore operation.

User Action: Mount the disk n on the drive ddnn.

BAC—Mount tape n on ddnn:

Explanation: BACKUP is requesting a tape for a backup or restore operation. This message prints every two minutes until the tape is loaded.

User Action: Mount the tape n on the drive ddnn and press RETURN.

BAC—Please answer yes or no

Explanation: BACKUP requires a YES or NO response.

User Action: Enter YES or NO at your terminal.

BAC—Starting disk n on ddnn:

Explanation: The disk n is being copied to (or from) the drive ddnn.

User Action: None.

BAC—Starting tape n on ddnn:

Explanation: The tape n is being copied to (or from) the drive ddnn.

User Action: None.

BAC—Starting verify pass

Explanation: The verify pass of a disk-to-disk operation is beginning.

User Action: None.

BAC—Starting verify pass tape n on ddnn:

Explanation: The tape n is being verified during a backup or restore operation.

User Action: None.

5.12.7.2 Warning Messages

BAC—•WARNING•—Allocation failure [directory]filename.type;version

Explanation: During a copy to a mounted volume, there was not enough free space to copy the specified file.

User Action: Delete unnecessary files to create free space on the volume. Then, reenter the command line.

BAC—•WARNING•—Appending at default bpi on ddnn:
BAC – •WARNING•—Appending at 1600 bpi on ddnn:

Explanation: Either of these messages indicates that you specified the wrong tape density with the BACKUP qualifier /APPEND. BACKUP performs an append operation only at the density at which the tape was previously written. The default bpi in the first message is either 800 or 6250, depending on the type of tape drive.

User Action: None. BACKUP continues at the density at which the tape was previously written.

BAC—•WARNING•—Bad block data error

Explanation: You entered an incorrect bad block location, count, or syntax.

User Action: For information on entering bad block information manually, see the description of the /BAD_BLOCKS:MANUAL qualifier. Then, enter the correct information.

BAC—•WARNING•—Block exceeds volume size

Explanation: You specified a bad block that is larger than the size of the output disk.

User Action: Enter the correct block.

BACKUP

BAC—•WARNING•—Boot block is bad

Explanation: The boot block is marked as bad on the output disk, so the disk is not hardware bootable.

User Action: If you need a hardware-bootable disk, wait until the operation completes. Then, repeat the procedure with a disk that has a usable boot block (logical block number 0).

BAC—•WARNING•—Boot block is corrupt

Explanation: The input disk does not contain a valid boot block, so the output disk will not be hardware bootable.

User Action: If you need hardware-bootable output, use an input disk with a valid boot block (LBN 0).

BAC—•WARNING•—Boot block read error

Explanation: An error occurred while BACKUP was reading the boot block. Although the input disk may contain a valid boot block, BACKUP was unable to read it. The output disk will not be hardware bootable.

User Action: If you need a hardware-bootable disk, wait until the operation completes. Then, repeat the procedure.

BAC—•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: If you need a hardware-bootable disk, wait until the operation completes. Then, repeat the procedure.

BAC—•WARNING•—Cannot restore contiguously
[directory]filename.type;version

Explanation: The output device does not contain enough contiguous blocks to restore the indicated file contiguously. The file will be restored noncontiguously.

User Action: To make the file contiguous again, use the DCL command COPY /CONTIGUOUS.

BAC—•WARNING•—Close or write attributes error
[directory]filename.type;version
IO Error code number

Explanation: During a copy to a mounted volume, BACKUP encountered an error while attempting to close the specified file.

User Action: Determine the cause of the error from the I/O code (see the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual*). If it is correctable, delete the portion of the file that BACKUP has copied. Then, reenter the command line.

BAC—•WARNING•—Data ID record verify error

Explanation: An error occurred while BACKUP was verifying an 80-byte data-ID record. (A data-ID record is a control record used by BACKUP.)

User Action: When the operation completes, use the BACKUP/COMPARE command to verify the data. If the compare operation fails, repeat the backup or restore operation.

BAC—•WARNING•—Data record verify error
[directory]filename.type;version
file ID number LBN number

Explanation: There is a difference in a data block on the 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, it is the owner directory from the file's header, not the directory in which the file is contained.

User Action: Repeat the backup operation. If it fails again, repeat the operation with a different disk or tape.

BAC—•WARNING•—Data was lost due to IO errors
[directory]filename.type;version

Explanation: A read error resulted in missing data. The files may contain invalid data, and some of the files may not be restored.

User Action: Repeat the backup operation.

BAC—•WARNING•—Directory verify error

Explanation: A directory record on the input device did not match a directory record on the output device.

This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting and error logging). Then, reenter the command line.

BAC—•WARNING•—Disk out of sequence. Please mount correct disk.

Explanation: You mounted the wrong volume on the disk drive during a restore operation from a multidisk backup set.

User Action: Mount the correct disk on the drive.

BAC—•WARNING•—Disk read error

Explanation: An unrecoverable read error occurred on the output disk, possibly caused by an undetected bad block, or an error occurred while BACKUP was sizing the input or output disk.

User Action: Use the ANALYZE/MEDIA command to locate all bad blocks on the output disk. Then, reenter the BACKUP command line. For information on the ANALYZE/MEDIA command, see Section 5.3.

BAC—•WARNING•—Duplicate blocks found

Explanation: You entered a bad block that is already listed in the bad block file.

User Action: None.

BAC—•WARNING•—EOT marker error

Explanation: During a backup operation, an error occurred while BACKUP was writing or verifying the end-of-tape (EOT) 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: Your response depends on the type of error that has occurred, as follows:

- On a write error, BACKUP rewinds the current tape and places it off line. BACKUP then asks you to mount a new tape. After you mount the new tape, BACKUP rewrites the data.
- 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 to the beginning-of-tape (BOT).

BAC—•WARNING•—Error accessing file
IO error code number
File ID number

Explanation: During a copy to a mounted volume, 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 continues with the next file.

User Action: For a definition of the error code number, see the I/O error codes in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.*

After BACKUP finishes, delete the file. Then, reenter the command line and specify the file on which the error occurred.

BAC—•WARNING•—Error accessing UFD Skipping [directory]
IO Error code number

Explanation: During a copy to a mounted volume, an error occurred when BACKUP attempted to access a directory.

User Action: For a definition of the error code number, see the I/O error codes in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.* Determine the cause of the error from the I/O error code. If the error is correctable, try the copy operation again.

BAC—•WARNING•—Error reading data blocks
IO Error code number
file ID number LBN number
RECOVERED

Explanation: An I/O error occurred while BACKUP was reading a data block from the input disk. The file identification (ID) of the file that contains the block and the logical block number (LBN) of the block are displayed, as well as the I/O error code.

If RECOVERED is printed after the message, BACKUP recovered the block by reading the disk again.

BACKUP continues the operation.

User Action: For a definition of the error code number, see the I/O error codes in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.*

BAC—•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: After the BACKUP operation completes, reenter the command line. Specify the directory indicated in the error message.

BAC—•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 and specify the directory indicated in the error message. If the error still occurs, use the File Structure Verification Utility (VFY) to find the lost files. For more information on VFY, see the *RSX-11M-PLUS Utilities Manual.*

BAC—•WARNING•—Extending index file

Explanation: The initial number of file headers is too small. If the number of blocks on the output disk is greater than or equal to 5120_{10}, BACKUP allocates 256_{10} additional headers. If the number of blocks on the output disk is less than 5120_{10}, BACKUP allocates 16_{10} additional headers.

User Action: None. BACKUP continues the operation with the extended file index.

BAC—•WARNING•—File header read error
[directory]filename.type;version
IO error code number

Explanation: An I/O error occurred while BACKUP was reading a file header, so the corresponding file was skipped. BACKUP continues with the next file.

User Action: For a definition of the error code number, see the I/O error codes in the *RSX–11M–PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual.*

BAC—•WARNING•—File header file verify error
[directory]filename.type;version

Explanation: A file header is not the same on the input and output devices. This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting and error logging). Then, reenter the command line.

BAC—•WARNING•—File ID area verify error

Explanation: The BACKUP-generated file identification (file ID) area of a data record is not the same on the input and output devices.

User Action: Refer to the previous message, "File header file verify error."

BAC—•WARNING•—File ID sequence number error
[directory]filename.type;version

Explanation: The two possible sources of this error are as follows:

- The sequence number in the file ID of a file does not match the sequence number of the file's entry in the directory.
- The sequence number of a directory does not match the sequence number of the directory's entry in the Master File Directory (MFD).

Therefore, the file or directory is not valid and is not copied.

User Action: Use the File Structure Verification Utility (VFY) to recover the lost files. For more information on VFY, see the *RSX–11M–PLUS Utilities Manual.*

BAC—•WARNING•—File IDs will not be preserved

Explanation: File IDs cannot be preserved because the index file bit map on the output device is too small. The value you specified with the /MAXIMUM_FILES qualifier is too small.

User Action: None. BACKUP continues the operation without preserving file IDs. If you want the disk to be hardware bootable, perform the BACKUP operation again, specifying a larger value with the /MAXIMUM_FILES qualifier.

BAC—•WARNING•—File marked for delete
[directory]filename.type;version

Explanation: The marked-for-delete bit (SC.MDL) of the system-controlled characteristics in the file header is set, indicating that the file is partially deleted. So, BACKUP does not copy the file.

User Action: None. BACKUP continues the operation with the next file.

BAC—•WARNING•—File not found [directory]filename.type;version

Explanation: During a backup operation, BACKUP cannot find a header in the index file (INDEXF.SYS) for a file or directory. BACKUP does not copy the file.

During the verify or compare pass of a restore operation, BACKUP cannot find the file on the output device.

User Action: None. BACKUP continues the operation with the next file.

BAC—•WARNING•—File not superseded
[directory]filename.type;version

Explanation: You attempted to restore a file to a mounted volume using the /NOSUPERSEDE qualifier (the default). The specified file was not restored because it already existed on the output disk.

User Action: If you want the file to be restored, reenter the command line. Specify the file and either the /SUPERSEDE or the /NEW_VERSION qualifier.

BAC—•WARNING•—Header ID record verify error

Explanation: The header-ID record generated by BACKUP on the output device is incorrect.

This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting and error logging). Then, reenter the command line.

BAC—•WARNING•—Header read error
[directory]filename.type;version
IO error code number

Explanation: During a backup operation, an I/O error occurred while BACKUP was reading a file header in the index file.

During a restore operation, this error is fatal; the operation terminates.

User Action: For a definition of the I/O error code number, see the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.*

BACKUP

BAC—•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 you must repeat the operation to ensure accuracy.

BAC—•WARNING•—Input volume structure level differs from output volume

Explanation: You initialized the output volume with a value for the /MAXIMUM_FILES qualifier that is greater than 25,593. As a result, the index file (INDEXF.SYS) on the output volume has more than one file header.

User Action: To make the volume structure the same on the input and output devices, initialize the output volume with a value for the /MAXIMUM_FILES qualifier that is less than 25,593.

BAC—•WARNING•—Internal error

Explanation: BACKUP has detected an error within itself.

User Action: Reenter the command line. If the message is repeated, the problem is in your software.

BAC—•WARNING•—Invalid date or time [directory]filename.type;version

Explanation: BACKUP encountered an invalid date or time in a file header during an incremental backup.

User Action: None. BACKUP continues the operation and the file is copied.

BAC—•WARNING•—Invalid disk format

Explanation: If you were performing an /IMAGE:RESTORE operation, you mounted a disk that is not a multivolume backup disk. If you were performing an /IMAGE:SAVE operation, you attempted to append to a disk that is not a multivolume backup disk. The disk may be corrupted.

User Action: Mount the correct disk and reenter the BACKUP command line.

BAC—•WARNING•—MFD read error

Explanation: An I/O error occurred while BACKUP was reading a block of the Master File Directory (MFD). BACKUP cannot copy the directories or files in that block of the MFD.

User Action: Reenter the command line. If the header still cannot be read, the files on the disk are lost. Use the File Structure Verification Utility (VFY) to recover the lost files. For more information on VFY, see the *RSX-11M-PLUS Utilities Manual*.

BAC—•WARNING•—No bad block data found

Explanation: You did not use the ANALYZE/MEDIA command to produce a file of the output disk's bad blocks. BACKUP continues the operation.

User Action: For more information on bad block processing by ANALYZE/MEDIA, see Section 5.3.

BAC—•WARNING•—No files found

Explanation: During a backup or restore operation, BACKUP did not find any files to transfer. You may have entered the wrong file or directory specification.

User Action: If you are sure you entered the correct file and directory specifications, no further action is necessary. If not, reenter the command, using the correct file specification.

BAC—•WARNING•—Nonfatal qualifier conflicts being ignored

Explanation: You entered a qualifier that conflicts with the rest of the command line. Specifically, you used the /REWIND, /DENSITY, or /LENGTH qualifier during a disk-to-disk copy operation, or you used the /NEW_VERSION, /SUPERSEDE, or /NOSUPERSEDE qualifier during a disk-to-tape backup operation.

User Action: None. BACKUP ignores all the conflicting qualifiers for the operation.

BAC—•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. However, there is no corresponding directory on the output volume.

User Action: Reenter the command line, specifying the /DIRECTORY qualifier to create the directory.

BAC—•WARNING•—Open error [directory]filename.type;version
IO error code number
File I/D number

Explanation: During a copy operation to a mounted volume, an error occurred while BACKUP was attempting to open the specified file.

User Action: For a definition of the error code number, see the I/O error codes in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.* If the error is correctable, delete any portion of the file already copied by BACKUP. Then, reenter the command line.

BAC—•WARNING•—Privilege violation [directory]filename.type;version

Explanation: During a backup operation, you attempted to copy a file that you did not have read access to.

User Action: None. BACKUP does not copy the file.

BAC—•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

BAC—•WARNING•—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.

BAC—•WARNING•—Tape label error on ddnn
IO Error code number

Explanation: An I/O error occurred while BACKUP was reading or writing a tape label. A read error is not fatal as long as BACKUP can continue reading the tape. If a write error has occurred, BACKUP will display the message "BAC—Mount another tape."

User Action: Reenter the command line, specifying a different tape.

BAC—•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.

This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting or error logging). Then, reenter the command line.

BAC—•WARNING•—Tape out of sequence. Please mount the correct tape.

Explanation: You mounted the wrong tape volume on the tape drive during a restore operation from a tape backup set.

User Action: Mount the correct tape on the drive.

BAC—•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, using /REWIND before attempting the append operation.

BAC—•WARNING•—Tape read error

Explanation: A nonrecoverable I/O error occurred while BACKUP was reading a tape. BACKUP ignores the error.

User Action: None.

BAC—•WARNING•—Tape write error
IO 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.

BAC—•WARNING•—This disk will not contain a hardware-bootable system

Explanation: The output disk will not be hardware bootable. This can be caused by one of the following conditions:

- The input disk is not bootable.
- The input disk is not the same size or type as the output disk.
- The system image (RSX11M.SYS) was not copied.

This message is not issued when BACKUP is restoring to a mounted volume.

User Action: If you want the output disk to be hardware bootable, use a bootable input disk and an output disk that is the same size as the input disk. Also, be sure the system image is included in the operation.

BAC—•WARNING•—UFD record verify error

Explanation: A directory record is not the same on the input and output devices.

This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting and error logging). Then, reenter the command line.

BAC—•WARNING•—VBN 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 disk.

User Action: None.

BAC—•WARNING•—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. Then, reenter the command line.

If you have the correct tape, reenter the command line with the /REWIND qualifier to position the tape.

BAC—•WARNING•—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, press the Write Enable switch on the disk drive.

BAC—•WARNING•—Wrong backup set

Explanation: During a restore operation from a multivolume backup set, BACKUP found that one of the tapes or disks did not contain the correct backup set.

User Action: Reenter the command line, specifying the correct tapes or disks.

BAC—•WARNING•—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.

5.12.7.3 Fatal Messages

BAC—•FATAL•—Allocation for system file exceeds volume limit

Explanation: One of the five system files that are created when a volume is initialized does not fit on the output disk. This message usually occurs when the output disk is smaller than the input disk.

User Action: Use the /MAXIMUM_FILES and /HEADERS qualifiers to reduce the number of headers indexed by the file INDEXF.SYS. You can also force the allocation to start at the beginning of the disk by reinitializing the disk with the BACKUP or INITIALIZE command and the /INDEX:BEGINNING qualifier.

BAC—•FATAL•—Ambiguous option

Explanation: You specified a BACKUP option that is not unique. For example, the "B" in /POSITION:B could mean either BEGINNING or BLOCK.

User Action: Use at least two characters to specify a unique form of a BACKUP option.

BAC—•FATAL•—Ambiguous qualifier

Explanation: You specified a BACKUP qualifier that is not unique. For example, /RE could mean either /REVISED or /REWIND.

User Action: Use at least three characters to specify a unique form of a BACKUP qualifier.

BAC—•FATAL•—Attach failed on ddnn:

Explanation: You specified a device that BACKUP cannot access.

User Action: Use the SHOW DEVICES command to see if the device is in use (allocated or mounted) by another user. Allocate an available device and repeat the operation.

BAC—•FATAL•—Bad block data error

Explanation: An error occurred reading the bad block information on the output disk.

User Action: Use the ANALYZE/MEDIA command to recreate the bad block information or use another disk.

BAC—•FATAL•—Bad block file full

Explanation: You entered more than 204_{10} sets of contiguous bad blocks. You cannot enter more bad blocks than the bad block file (BADBLK.SYS) can hold.

User Action: If the number of contiguous bad blocks on your disk exceeds 204_{10}, you can no longer use it effectively. Use another disk.

BAC- •FATAL•—Cannot append on continuation volume

Explanation: The /APPEND qualifier was specified with a continuation volume. A backup set can be appended to the first volume only.

User Action: Mount a blank volume and reenter the command line without the /APPEND qualifier.

BAC—•FATAL•—Close or write attributes error
[directory]filename.type;version
IO Error code number

Explanation: During a copy to a mounted volume, BACKUP encountered an error while attempting to close the specified file.

User Action: Determine the cause of the error from the I/O code (see the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual*). If it is correctable, delete the portion of the file that BACKUP has copied. Then, reenter the command line.

BAC—•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 without one of the mutually exclusive qualifiers.

BAC—•FATAL•—Device conflict

Explanation: You specified both tape and disk drives as part of either the input device specification (for a restore operation) or the output device specification (for a backup operation).

User Action: Reenter the command line, specifying a single type of device (that is, either a tape or a disk drive).

BAC—•FATAL•—Device not in system

Explanation: A device was specified that is not known to the system.

User Action: To obtain a list of the devices currently available, use the SHOW DEVICES command. Reenter the command line, specifying a device that is known to the system.

BAC—•FATAL•—Device not mounted Files–11 on ddnn:

Explanation: The BACKUP qualifiers specified on the command line cannot be used unless the device is mounted with the Files–11 format.

User Action: Mount the device Files–11 and reenter the command line, or use a different command line. (For a list of the correct combinations of mounted devices and qualifiers, see Table 5–3.)

BAC—•FATAL•—Device not mounted foreign on ddnn:

Explanation: The BACKUP qualifiers specified on the command line cannot be used unless the device is mounted foreign.

User Action: Mount the device foreign and reenter the command line, or use a different command line. (For a list of the correct combinations of mounted devices and qualifiers, see Table 5–3.)

BAC—•FATAL•—Device not supported

Explanation: You specified a device that is not supported by the operating system.

User Action: Check the command line for valid device names. Then, correct any errors and reenter the command.

BAC—•FATAL•—Directive error

Explanation: An internal error occurred in BACKUP. This may indicate that pool is low.

User Action: Reenter the command line. If the message is repeated, the problem may be in your software.

BAC—•FATAL•—Disk read error

Explanation: An unrecoverable read error occurred on the output disk, possibly caused by an undetected bad block, or an error occurred while BACKUP was sizing the input or output disk.

User Action: Use the ANALYZE/MEDIA command to locate all bad blocks on the output disk. Then, reenter the BACKUP command line. For information on the ANALYZE/MEDIA command, see Section 5.3.

BAC—•FATAL•—Disk write error

Explanation: An unrecoverable write error occurred on the output disk. The error could have been caused by an undetected bad block.

User Action: Use the ANALYZE/MEDIA command to locate all bad blocks on the output disk. Then, reenter the BACKUP command line. For information on the ANALYZE/MEDIA command, see Section 5.3.

BAC—•FATAL•—Doubly defined qualifier

Explanation: You specified the same qualifier more than once on the command line.

User Action: Reenter the command line, specifying the qualifier only once.

BAC—•FATAL•—End of volume encountered. Backup set not found.

Explanation: You specified a backup set that is not on the tape or disk volume.

User Action: Mount the correct tape or disk volume. If you have the correct volume mounted, reenter the command line and specify the correct backup set name.

BAC—•FATAL•—Error limit exceeded

Explanation: BACKUP reached the maximum number of read errors allowed and terminated execution.

User Action: Using a different tape drive, reenter the command line. If you do not have another drive, clean the tape drive heads on the original drive. Then, reenter the command line.

BAC—•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.

BAC—•FATAL•—Failed to read bad block file

Explanation: BACKUP was unable to read the bad block information from a last-track output disk.

User Action: Use the ANALYZE/MEDIA command with the /OVERRIDE qualifier to locate all bad blocks. Then reenter the BACKUP command line, using the /BAD_BLOCKS:OVERRIDE qualifier.

BAC—•FATAL•—File ID exceeds maximum number of files

Explanation: You specified a maximum number of files with the /MAXIMUM_FILES 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_FILES qualifier.

BAC—•FATAL•—File not found

Explanation: You specified an indirect command file or backup set file that BACKUP could not find.

User Action: Reenter the command line and specify the correct file name.

BAC—•FATAL•—Handler not resident

Explanation: You specified a device that does not have a driver loaded in memory.

User Action: Load the driver for the specified device. If the driver you intended to use is currently loaded, then reenter the command line and specify the correct device name.

BAC—•FATAL•—Header read error [directory]filename.type;version IO Error code number

Explanation: During a restore operation, an I/O error occurred while BACKUP was reading a file header in the index file. The operation terminates.

User Action: For a definition of the error code number, see the *RSX–11M–PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual.*

BAC—•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: Before initializing the disk, use the ANALYZE/MEDIA command to find all the bad blocks. Then, reenter the BACKUP command line.

BAC—•FATAL•—Illegal use of directory qualifier

Explanation: You specified one of the following:

- The /LIST qualifier with a disk or tape that is not part of a backup set
- The /LIST qualifier with the /INITIALIZE qualifier (or one of its related qualifiers)

User Action: If you are initializing a device, do not use the /LIST qualifier.

BAC—•FATAL•—Inconsistent /INITIALIZE qualifiers

Explanation: You specified the /NOINITIALIZE qualifier with one or more of the initialization qualifiers for the output disk.

For a list of the initialization qualifiers, see Section 5.12.5.1.

User Action: Reenter the /NOINITIALIZE command line without using any of the initialization qualifiers.

BAC—•FATAL•—Index file header read error IO error code number

Explanation: An I/O error occurred while BACKUP was reading the header of the index file on the input disk.

User Action: For a definition of the error code number, see the *RSX–11M–PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual.* Reenter the command line.

BAC—•FATAL•—Index file write error

Explanation: An I/O error occurred while BACKUP was writing the index file on the output disk.

User Action: Use the ANALYZE/MEDIA command (see Section 5.3) to identify the bad blocks on the output disk; then, reenter the command line.

BAC—•FATAL•—INDEXF.SYS is full

Explanation: The index file, INDEXF.SYS, cannot map any more file headers.

User Action: Reenter the command line, specifying larger values for the /MAXIMUM_FILES and /HEADER qualifiers.

BAC—•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.

BAC—•FATAL•—Input line too long

Explanation: The maximum length of a command line is 256_{10} 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.

BAC—•FATAL•—Internal error

Explanation: BACKUP has detected an error within itself.

User Action: Reenter the command line. If the problem recurs, there is a problem with your software.

BAC—•FATAL•—Invalid date or time

Explanation: You specified an incorrect date or time in the command line.

User Action: Specify the correct date or time.

BAC—•FATAL•—Invalid density

Explanation: This error occurs for one of the following reasons:

- You specified a density tape that was not 1600, 800, or 6250 bpi.
- You attempted to use both 7-track and 9-track tapes in a multivolume tape set.

User Action: If you specified the wrong density, reenter the command line with the correct density. For a multivolume tape set, use one type of tape (7-track or 9-track).

BAC—•FATAL•—Invalid disk format

Explanation: If you were performing an /IMAGE:RESTORE operation, you mounted a disk that is not a multivolume backup disk. If you were performing an /IMAGE:SAVE operation, you attempted to append to a disk that is not a multivolume backup disk.

User Action: Mount the correct disk and reenter the BACKUP command line.

BAC—•FATAL•—Invalid tape format

Explanation: An invalid tape record was read during a restore operation.

User Action: None. The invalid record is not restored.

BACKUP

BAC—•FATAL•—Invalid filename

Explanation: The name of the indirect command file is not syntactically correct.

User Action: Reenter the command line with the correct file name.

BAC—•FATAL•—Invalid value or name

Explanation: A value or name specified for a qualifier has incorrect syntax or is out of range.

User Action: Refer to the description of the qualifier to determine the correct values for it.

BAC—•FATAL•—Manufacturer bad sector file is corrupt

Explanation: BACKUP was unable to read the bad block information from a last-track output disk.

User Action: Using the ANALYZE/MEDIA command with the /OVERRIDE qualifier, check the disk for bad blocks. Then reenter the command line, specifying the /BAD_ BLOCKS:OVERRIDE qualifier.

BAC—•FATAL•—MFD read error

Explanation: An I/O error occurred while BACKUP was reading a block of the Master File Directory. BACKUP cannot copy the directories or files in that block of the MFD.

User Action: Reenter the command line. If the header still cannot be read, the files on the disk are lost. Use the File Verification Utility (VFY) to recover the lost files. For more information on VFY, see the *RSX-11M-PLUS Utilities Manual.*

BAC—•FATAL•—Missing colon

Explanation: A qualifier that accepts an option was not followed by a colon.

User Action: Reenter the command line and include the missing colon and value.

BAC—•FATAL•—More than 1 level of indirection

Explanation: BACKUP does not support more than one level of indirect command files; you cannot nest indirect command files when you are using BACKUP.

User Action: Reenter the command line, specifying an indirect command file that does not use other command files.

BAC—•FATAL•—Name exceeds maximum allowed length

Explanation: You specified a name (such as a backup set name) that is longer than 12 characters.

User Action: Specify a name that is shorter than 12 characters.

BAC—•FATAL•—Number of headers inconsistent with maximum files

Explanation: During an attempt to initialize an output volume, you specified a maximum number of files (using the /MAXIMUM_FILES qualifier) that is inconsistent with the number of headers specified with the /HEADERS qualifier.

User Action: For the valid ranges of values for the /MAXIMUM_FILES and /HEADERS qualifiers, see the description of the INITIALIZE command in Section 5.12.5.1.

BAC—•FATAL•—Open error [directory]filename.type;version
IO error code number
File I/D number

Explanation: During a copy operation to a mounted volume, an error occurred while BACKUP was attempting to open the specified file.

User Action: For a definition of the error code number, see the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual* or the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual.* If the error is correctable, delete any portion of the file already copied by BACKUP. Then, reenter the command line.

BAC—•FATAL•—Output disk too fragmented to restore

Explanation: The internal tables in BACKUP have overflowed because of extreme fragmentation of the output disk. If the output disk 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 as the output device.

BAC—•FATAL•—Output device is full

Explanation: There are no free blocks on the output disk. This can occur when the output disk is smaller than the input disk or during an append to a tape or disk that is already full.

User Action: Use another output disk or tape with more free space, or reenter the command line, specifying a subset of the files.

BAC—•FATAL•—Override invalid with non-last-track device

Explanation: The /BAD_BLOCKS:OVERRIDE qualifier is valid only when the output disk is a last-track device.

User Action: For information on the OVERRIDE option, see Section 5.12.5.1.

BAC—•FATAL—Privilege violation

Explanation: The mount status of one of the devices is inconsistent with the qualifiers specified in the command line. For example, the device may be mounted foreign, but the qualifiers you used apply to Files-11 devices only.

User Action: Use the SHOW DEVICES command to check the mount status of the device. Then reenter the command line with the appropriate qualifiers (see Table 5-3).

BAC—•FATAL•—Ran out of spare file IDs

Explanation: The output disk required more file headers than the input disk, but no free headers were available. The lack of headers is probably due to one of the following two reasons:

- The output disk is too fragmented because of bad blocks.
- There are no free file headers on the input disk.

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 is too fragmented, use the ANALYZE/MEDIA command (see Section 5.3) to display the number of bad blocks. If it contains a large number of bad blocks, use a different disk.
- Use the DCL command DIRECTORY/FREE to display the number of free file headers on the input disk. If there are fewer than four free headers, delete some of the files and reenter the command line. If you still do not have enough file headers, specify the /NOPRESERVE qualifier in the command line.

BAC—•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.

BAC—•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.

BAC—•FATAL•—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.

BAC—•FATAL•—Search for home block failed

Explanation: The home block could not be found on the input disk. Either the home block is bad or the disk is not in Files–11 format.

User Action: Check to see that you have the correct disk.

BAC—•FATAL•—Stack overflow in sort routine

Explanation: BACKUP has detected an error within itself.

User Action: Reenter the command line. If the problem recurs, your software may be inoperable.

BAC—•FATAL•—Syntax error

Explanation: The command line is invalid.

User Action: Reenter the command line, using valid syntax.

BAC—•FATAL•—Tape label error on ddnn
IO Error code number

Explanation: An I/O error occurred while BACKUP was writing a tape label on the first tape of an /APPEND operation.

User Action: Reenter the command line, specifying a different tape.

BAC—•FATAL•—Tape not at BOT. No rewind or append specified.

Explanation: For a backup operation to tape, unless you specify the /APPEND qualifier BACKUP does not process a tape that is not at the beginning of the tape (BOT).

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.

BAC—•FATAL•—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, using /REWIND before attempting the append operation.

BAC—•FATAL•—Tape positioning error. No EOV encountered.

Explanation: The tape-spacing operation failed to find the end-of-tape (EOT) volume for an append operation.

User Action: Reenter the command line.

BAC—•FATAL•—Tape read error

Explanation: A nonrecoverable I/O error occurred while BACKUP was reading a tape. BACKUP terminates the operation.

User Action: None.

BACKUP

BAC—•FATAL•—Tape to tape not supported

Explanation: BACKUP does not back up a tape to another tape.

User Action: Use a fixed disk to perform a backup operation.

BAC—•FATAL•—Too many devices

Explanation: You may specify only one disk drive as the input device for a backup operation and as the output device for a restore operation.

You may specify up to eight tape drives as the output devices for a backup operation and as the input devices for a restore operation. If you use the /IMAGE qualifier, you may specify up to eight disk drives as the output devices for a backup operation (/IMAGE:SAVE) and as the input devices for a restore operation (/IMAGE:RESTORE).

In a disk-to-disk copy operation, you may specify only one disk drive as the input device and only one disk drive as the output device.

User Action: Reenter the command line, specifying the correct number of drives.

BAC—•FATAL•—Too many file specifications

Explanation: You specified more than 16 file specifications on the command line.

User Action: Reenter the command line. To reduce the number of file specifications on the command line, use wildcards.

BAC—•FATAL•—UFD or MFD requires unsupported extension headers

Explanation: BACKUP does not support extension headers for Master File or User File Directories (MFDs or UFDs, respectively).

User Action: Reenter the command line. If the problem recurs, your software may be inoperable.

BAC—•FATAL•—Unknown option

Explanation: You specified an option that BACKUP does not recognize.

User Action: Check the qualifiers and options, then reenter the command line.

BAC—•FATAL•—Unknown qualifier

Explanation: You specified a qualifier that BACKUP does not recognize.

User Action: Check the qualifiers and reenter the command line.

BAC—•FATAL•—Unsupported structure level

Explanation: The file structure level on the input disk is not supported by BACKUP.

User Action: Check to be sure you have placed the correct disk in the drive.

For more information on file structures, see the descriptions of the /HEADERS and /MAXIMUM_FILES qualifiers.

BAC—•FATAL•—VBN 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 disk.

User Action: None.

BAC—•FATAL•—Verify lost

Explanation: During the verify pass, BACKUP lost synchronization between the input and the output. Either the tape position was lost, or you backed up from a disk that was changed during the backup operation.

This error can occur when files are created, deleted, or renamed during the BACKUP operation; BACKUP is unable to verify the data on the output device. It can also indicate a hardware error.

User Action: Be sure the contents of the input disk do not change during the data transfer or verify operations. Before repeating the operation, be sure there is no activity on the disk. If you are backing up or restoring the system disk, be sure there is no activity on the system (such as Resource Accounting and error logging). Then reenter the command line.

BAC—•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. Then reenter the command line.

If you have the correct tape, reenter the command line with the /REWIND qualifier to position the tape.

BAC—•FATAL•—Volume not ready

Explanation: The device is not on line.

User Action: Put the device on line and reenter the command line.

BAC—•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, press the write-enable switch on the disk drive.

BAC—•FATAL•—Wrong backup set

Explanation: During a restore operation from a multivolume backup set, BACKUP found that one of the tapes or disks did not contain the correct backup set.

User Action: Reenter the command line, specifying the correct tapes or disks.

BAC—•FATAL•—Wrong input volume label

Explanation: The input volume label specified with the /LABEL:INPUT 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.

5.13.1 SHOW ASSIGNMENTS and SHOW LOGICALS

SHOW ASSIGNMENTS and SHOW LOGICALS display 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-PLUS and Micro/RSX 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.

/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 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.

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? /qualifier

SET DEVICE:ddnn:/qualifier[s]

Device Attributes

/[NO]CACHE:(arg[,s])
CREATE[:[region][:[mainpartition][:[size]]]]
REGION:name
[NO]DEFER_WRITES
[NO]DIRECTORY[:extentsize]
[NO]LOGICAL[:extentsize]
[NO]OVERLAY[:extentsize]
[NO]READ_AHEAD[:extentsize]
[NO]VIRTUAL[:extentsize]
/[NO]CHECKPOINT_FILE:n
/[NO]LOWERCASE
/[NO]PUBLIC
/[NO]SYSTEM
/WIDTH:n

Parameter

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 ddnn: is preceded by a colon (:).

Device Attributes

/[NO]CACHE:(arg[,s])
CREATE[:[region][:[main_partition][:[size]]]]
REGION:name
[NO]DEFER_WRITES
[NO]DIRECTORY[:extentsize]
[NO]LOGICAL[:extentsize]
[NO]OVERLAY[:extentsize]
[NO]READ_AHEAD[:extentsize]
[NO]VIRTUAL[:extentsize]

Associates data caching of disk requests with the specified disk and allows you to override the disk data caching defaults and set one or more of the options. The values to which the options are set with the SET DEVICE/CACHE command become the default values. Subsequent commands change only the options specified in the new command line.

The value of the extentsize argument in the options specifies the maximum size (in decimal disk blocks) of an I/O request that will be considered for caching. The minimum acceptable value is 1 block; the maximum is 127 blocks.

The default is /NOCACHE. Note that if you use only one option in a command line, you can omit the parentheses. For more information on data caching, see the *RSX-11M-PLUS and Micro/RSX System Management Guide*. The options are defined as follows:

CREATE[:[region][:[mainpar][:[size]]]]	Creates a cache region in memory and associates caching for the specified device with that region. You can specify the name of the region to be created, the main partition in which the region is to be created, and the size of the region. You specify the size of the region in decimal number of disk blocks. By default, the /CACHE qualifier uses a region called CACHE in the GEN partition with a size of 100_{10} disk blocks.
REGION=name	Associates caching for the specified device with an already existing cache region. The default region is CACHE.
[NO]DEFER_WRITES	Enables or disables data caching of deferred write requests. The NODEFER_WRITES option is the default.
[NO]DIRECTORY[=extentsize]	Enables or disables caching of directory and other volume-structured I/O; that is, logical I/O from an ACP (for example, F11ACP). The default is DIRECTORY. The default extent size for the DIRECTORY option is 1 block.
[NO]LOGICAL[=extentsize]	Enables or disables caching of logical I/O. The default is LOGICAL. The default extent size for the LOGICAL option is 1 block.
[NO]OVERLAY[=extentsize]	Enables or disables caching of overlay I/O. The default is OVERLAY. The default extent size for the OVERLAY option is 4 blocks.
[NO]READ_AHEAD[=extentsize]	Enables or disables reading of the next extent before reading is explicitly requested. The default is /[NO]READ_AHEAD. The default extent size for the READ_AHEAD option is 5 blocks.
[NO]VIRTUAL[extentsize]	Enables or disables caching of virtual I/O. The default is /VIRTUAL. The default extent size for the VIRTUAL option is 5 blocks.

SET DEVICE

/[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.

/[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 compatibility with the VMS operating system.

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_{10} characters on the line printer LP0:.

```
$ 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_{10} 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: The command attempted to set a pseudo device public.

User Action: A pseudo device cannot be set public. Reenter the command using the physical device name.

SET—Write check not supported

Explanation: The 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 on RL01s.

5.13.3 SHOW DEVICES

SHOW DEVICES displays information about devices recognized by the system. Note that you cannot specify both a device and a qualifier in the same command line.

The command SHOW DEVICES ddnn: is useful when you plan to allocate a drive, since it shows you which drives are available. Because the same controller may control devices whose disks are not interchangeable, the command ALLOCATE DU may allocate a device that cannot accept your disk.

Formats

SHOW DEVICES:ddnn:
SHOW DEVICES/qualifier[,s]

Parameter

ddnn:

Displays information about all devices of a particular type on the system. The 2-letter device mnemonic dd: is terminated with a colon and indicates the type of device controller. The display shows the devices under that type of controller by model name. The nn specifies the unit of the device about which you want to display information. If you do not specify a device, the system displays information about all devices known to the system, including terminals and pseudo devices.

Device Qualifiers

/[NO]CACHE
/[NO]PUBLIC
/WIDTH:ddnn:
/[NO]SYSTEM (synonym for /PUBLIC)

Device Qualifiers

/[NO]CACHE

Displays all cached or uncached devices. If you specify /CACHE and no devices are cached, the system does not display any information. For further information on data caching, see the description of the SHOW CACHE command in Section 5.13.4.

/[NO]PUBLIC
/[NO]SYSTEM

Displays a list of all devices set (or not set) public. The /[NO]SYSTEM qualifier is a synonym included for compatibility with the VMS operating system.

/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. For an example of a display, see the examples section following this list.

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 in 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 SHOW DEVICES/NOPUBLIC, the display is PUB=ddnn: or NOPUB=ddnn:.

TYPE=

Indicates the device type by model name, for example, RL02. 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 on the terminal.

LOADED

Indicates that a loadable device driver is currently loaded.

UNLOADED

Indicates that a loadable device driver is currently not loaded.

SPOOLED

Indicates that a device is under the control of the Queue Manager.

WCHK=
NOWCHK=

Indicates a device with write-checking enabled or not enabled.

SHOW DEVICES

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
C00:      TT0:
TT0:      LOADED
TT1:      [7,40]  [1,1]   - LOGGED ON LOADED
TT2:      [7,40]  [14,10] - LOGGED ON LOADED
TT3:      [303,5] [303,5] - LOGGED ON LOADED
                     .
                     .
                     .
TT66:     LOADED
TT67:     LOADED
NL0:      LOADED
VT0:      LOADED
VT1:      LOADED
VT2:      [1,1]   [1,1]   - LOGGED ON LOADED
VT4:      [303,5] [303,5] - LOGGED ON LOADED
TI0:
CL0:      TT0:
SP0:      DB0:
LB0:      DB0:
SY0:      DB0:
```

This example shows the display from SHOW DEVICES. All devices and pseudo devices are included. For terminals, the 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 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 (DUnn:) identifies the device controller, and the model name (RD51, RX50) 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:LPO:/WIDTH RET
BUF=LPO: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.

5.13.4 SHOW CACHE

SHOW CACHE invokes the Resource Monitoring Display (RMD) to provide displays of detailed information on data caching. For more information, see the *RSX-11M-PLUS and Micro/RSX System Management Guide*.

Format

SHOW CACHE/DEVICE[:ddnn:][/qualifier[,s]]

Command Qualifiers

/DEVICE:ddnn:
/RATE:nn

Command Qualifiers

/DEVICE:ddnn:

Specifies the device about which you want to display detailed data caching information. If you do not specify a device, the system displays information about the system disk, SY:.

/RATE:nn

Specifies in seconds the rate at which the user's terminal screen refreshes while the caching information is being shown.

Chapter 6

The LINK and LIBRARY Commands

This chapter describes the process of developing a program on RSX–11M–PLUS. To create a source program, you must be familiar with editing. You must also know 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 the language.

If you are not familiar with the rudiments of this process, see the *Introduction to RSX–11M–PLUS.*

RSX–11M–PLUS systems offer extensive program development facilities. These facilities are introduced in the *RSX–11M–PLUS Guide to Program Development.* That manual introduces the most important software tools used in program development and describes the program-development process in considerable detail.

6.1 Linking a Task with 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 also can be built by the Task Builder in complex ways. This chapter describes the basic functions of the Task Builder and the LINK command. Experienced RSX programmers may want to read the *RSX–11M–PLUS and Micro/RSX Task Builder Manual* instead.

6.2 Review of the Programming Process

This section reviews the steps you must take in writing a program and creating a task. It also discusses the files that are created in this process.

6.2.1 Basic Steps in Programming

The programming process can be divided into several major steps. Here the discussion includes only the basic steps that you take to create a running task. Other steps, such as debugging and testing, are not discussed.

The three major steps are as follows:

1. Writing the program in a supported source language
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 in one of several source languages. A source language is a specific collection of defined symbols, coded phrases, and characters. The source language may resemble closely the binary machine language. If it does, it is called an assembly language. If the source language more closely resembles English (or another human language), 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, using an editor.

The second step in the programming process is assembling or compiling the program. If you wrote the code in assembly language, you use DIGITAL's MACRO–11 Relocatable Assembler to assemble it. 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 in 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. A task is also in a binary language readable by the system. However, a task differs from object code in that the task includes all the addresses of information to which it will refer, its own addresses, arranged in order, and special information for use by the system. You create a task by using the LINK command, which invokes the Task Builder. The Task Builder uses the object code in the object files that you created and transforms it into a runnable task.

The following sections contain brief descriptions 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 the file type MAC when you use the MACRO–11 assembly language and a file with the file type 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 a file called an object file.

6.2.3 The Object File

The object file, which has the OBJ file type, 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 display the object file on the screen, the file will appear in the same way 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 have the proper code for printing on the printer or displaying on the terminal. By itself, the object file cannot execute as a task. To create a task you must use the Task Builder.

6.2.4 The Task File

The task file, which has a TSK file type, 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 the following four types of input files:

- Object files
- Library files
- Overlay description files
- Command files

These files are described in the following sections.

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 multiply 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.
- Logically discrete parts make 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 shows how the LINK command is used to create a task from object files:

```
$ 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 has an OLB file type. The library file contains coded routines that may be used by more than one task. The tasks call the routines in the library file when they are needed, thus saving time and effort. These routines may include, for example, calculating routines such as addition or multiplication. The LIBRARY command is used for maintaining libraries (see Section 6.6).

A library file may be specified in the LINK command line in two ways. The first is shown in the example that follows. The two ways are fully described in a later section.

```
$ 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 discussed previously. It contains special language that contains the names of object files and libraries in an algebraic relationship. This file has the ODL file type.

This chapter does not describe overlays, 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-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 usable only 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 here. The following example shows how the LINK command is used to build a task from a command file.

```
$ 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. However, a large program can be coded and compiled in several discrete sections called modules, which must have unique names. 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 performed by the Task Builder, as follows:

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 the following:

```
01100011010110001110001010111010001010000111
```

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, invokes the Task Builder and creates a task from one or more object modules. The following is an example of a 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.

In Figure 6–1, three modules of equal size (2000 octal words) have been compiled. Each module has addresses starting at 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.

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 and a special output file while building the task.

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–PLUS and Micro/RSX Task Builder Manual.*

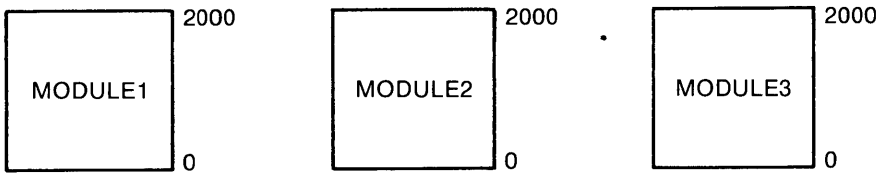
6.5 The LINK Command

The LINK command invokes the Task Builder, a system utility that builds the tasks, called task images, that run on the system.

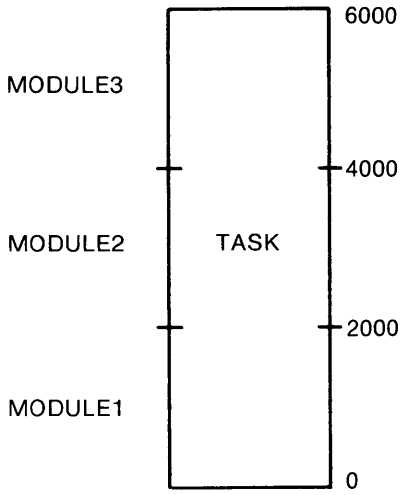
The unadorned LINK command includes a number of defaults that produce a standard task image file, having the file type TSK. The Task Builder (TKB), however, provides you with many choices that are not included in the defaults. Thus, you can use qualifiers to the LINK command to perform the following 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

Figure 6–1: Module Address Assignment



3 SEPARATE MODULES OF 2K-WORDS EACH



TASK OF 3 MODULES CREATED BY THE TASK BUILDER

ZK-1371-83

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 you want both a task image file and 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 send it to the line printer.

If you are linking more than one object module to form a task, the Task Builder, by default, names the output files after the first task named in the command line. For example, the following command:

```
$ LINK MOE, CURLY, LARRY [RET]
$
```

produces a task image file named MOE.TSK. That is, LINK uses the name of the first input file, MOE, as the name of the task unless you change the name of the task. Use the /TASK qualifier to change the name of the task image file. For instance, the following command:

```
$ LINK/TASK:SHEMP MOE, CURLY, LARRY [RET]
$
```

produces a task image file named SHEMP.TSK. Or, use the following command:

```
$ LINK MOE, CURLY/TASK, LARRY [RET]
$
```

to produce a task image file named CURLY.TSK.

Note, however, that the following LINK command:

```
$ LINK/MAP MOE, CURLY/TASK, LARRY [RET]
$
```

produces a task image file named CURLY.TSK and a map file named MOE.MAP because MOE is the first input file. If you use the following form:

```
$ LINK/MAP:SAM MOE,CURLY,LARRY [RET]
$
```

the map file is named SAM instead of MOE.

In addition, if 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]
$
```

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 take one of the following steps:

- 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.
- Specify options in a separate file. When prompted by **Option?**, simply name that file with a preceeding at sign (@) and then press RETURN.
- Specify options in a separate file and name that file as an argument to the /OPTIONS qualifier. Note, however, that you will not be prompted for options.

Here are three examples.

```
$ LINK/MAP/OPTIONS BIGJON [RET]
Option? UNITS=7 [RET]
Option? ASG=SY0:6:7 [RET]
Option? [RET]
$
```

or

```
$ LINK/MAP/OPTIONS BIGJON [RET]
Option? @SPARKY.CMD [RET]
Option? [RET]
$
```

or

```
$ LINK/MAP/OPTIONS:@SPARKY.CMD BIGJON [RET]
$
```

The LINK command and its qualifiers are all DCL commands, and the format is DCL format. The Task Builder, however, does not recognize DCL format. It recognizes only Task Builder format. The translation from DCL format to TKB format is performed by DCL.

Issuing the LINK command to DCL initiates the following process:

1. DCL reads the command line and checks it for correct DCL format.
2. DCL translates the command line into correct TKB format.
3. DCL places the TKB-format command in a TKB indirect command file and passes the indirect command file to the Task Builder.

4. The Task Builder builds the task as directed and creates all requested output files.

5. The Task Builder returns control to DCL and DCL deletes the TKB indirect command file.

If you are building standard tasks by using the defaults and by following the examples given in the command description (Section 6.5.1), this process will serve you well.

See the *RSX-11M-PLUS and Micro/RSX Task Builder Manual* for more 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 fully describes 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 consult 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 in which you name them determines which output files you receive. The Task Builder can produce up to three kinds of output files. See the examples that follow.

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 (=).

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.

You can invoke the Task Builder from DCL in four ways:

- Issuing the LINK command with DCL qualifiers
- Invoking a DCL indirect command file including a LINK command with DCL qualifiers, such as DCL> @LINKFILE
- Issuing the LINK command and calling a TKB indirect command file, such as DCL> LINK @TKBFILE
- 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
[NO]FPP
OTS_FAST
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]
/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 argument n specifies the base relocation register. Acceptable values are 0, 4, or 5. The default is 5.

In TKB format, apply the /AC:n switch to the TSK file specification.

/[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 options.

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
OTS_FAST
PIC
POSITION_INDEPENDENT

Specifies that the code for the task relies on certain hardware elements or employs certain techniques. See the *RSX-11M-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 you to use the fast-mapping facility 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 default is NOFPP.

The OTS_FAST argument enables you to use the fast-mapping facility of the Executive to map windows for autoloaded memory-resident overlays, thereby increasing the speed of overlay mapping by approximately 10 times.

The PIC argument specifies that the resident common library or the 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.

/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–PLUS and Micro/RSX Debugging Reference Manual* for further information.

If you have a user-written debugger, name the file it is in when you use this qualifier. This file should be an object module.

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.

/[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 list includes 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.

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-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 words in size 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.

/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.

/OPTION[:@optionsspec]

Specifies one or more Task Builder options. Use this qualifier if you need to use any of the Task Builder options. Information on these options is summarized in Table 6–1. See Section 6.5 for information on using this qualifier. For full information, or options, see the *RSX–11M–PLUS and Micro/RSX Task Builder Manual.*

You can supply options for this qualifier in two ways: supply the file specification of a file containing the options, or 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. For example:

```
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.

/POSTMORTEM

Requests a postmortem dump should the task be terminated abnormally. See the *RSX-11M-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 the 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 Message (SMSG$). 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 ensures 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 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 the 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 the system library. See also the /SEGREGATE qualifier discussion.

In TKB format, apply the /SQ switch to the TSK file specification.

/SHAREABLE[:arg]

COMMON
LIBRARY
TASK

Informs the Task Builder that one of three things is happening depending on the argument applied to this qualifier:

- 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.

- 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.
- 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.

/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 argument 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 (the system library) 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 the system library 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-PLUS and Micro/RSX Task Builder Manual* for more information.

In TKB format, apply the /MA switch to the MAP file specification.

/[NO]TASK[:taskspec]

Specifies a name for the task image file different from that of the first input file plus the type TSK. If used as a file specification qualifier, the task name is derived from the name of the file to which the qualifier is attached.

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 are suppressed:

```
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.

/[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 Qualifiers

Any input file can have a parameter qualifier applied to it, identifying the kind of file that it is.

/[NO]CONCATENATE

Identifies the input file as a concatenated object file; this is the default. All modules in the file are processed to form the task image. 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.

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.

Table 6–1: Task Builder Options

Description and Language Key [1]	Format and Defaults [2]
	Control
Direct TKB to terminate build (M,H)	ABORT=0. Number is used for format only. Any number will do.
	Identification
Declare task name (M,H)	TASK=1–6 alphanumeric characters TASK=first 6 characters of task image file name
Set task's User Identification Code (M,H)	UIC=[g,m] UIC=terminal UIC
Declare task's priority (M,H)	PRI=decimal integer 1–250; PRI=set by INSTALL; 50 for nonprivileged users
Declare task's partition name and size (M,H)	PAR=parname[:base:length] PAR=GEN
Declare 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
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 processing format strings (H)	FMTBUF=decimal integer for length of format buffer (larger than default) FMTBUF=132
Declare extension of program section (M,H)	EXTSCT=psectname:extension. No default.
Declare extension of task's portion of memory; extends D-space portion of I- and D-space tasks (M,H)	EXTTSK=decimal integer for increase in task memory allocation (in words). Default from PAR option.
Declare virtual base address and size of virtual program section (H)	VSECT=psectname:base:window[:length]. No default.

[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 one exists. This chart is intended only as a reminder of the options, their formats, and their defaults. Consult the *RSX-11M-PLUS and Micro/RSX Task Builder Manual* for full information on these options.

Table 6–1 (Cont.): Task Builder Options

Description and Language Key [1]	Format and Defaults [2]
Declare number of additional window blocks needed by task (M,H)	WNDWS=1–23
Declare size of stack (M,H)	STACK=decimal integer for number of words needed for stack STACK=256
Sharing Storage	
Declare intention to access system-owned common block for data in memory; if used with I- and D-space tasks, common mapped with D-space APRs only (M,H)	COMMON=name:accesscode[:apr]. No default.
Declare intention to access system-owned resident library for code; if used with I- and D-space tasks, library mapped with I- and D-space APRs (M,H)	LIBR=name:accesscode[:apr]. No default.
Declare intention to access user-owned common block; if used with I- and D-space tasks, common mapped with D-space APRs only (M,H)	RESCOM=filespec/accesscode[: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/accesscode[:apr]. Standard defaults on filespec.
Declare intention to access a user-owned supervisor-mode library (M,H)	RESSUP=filespec/[-]SV[:apr]. Standard defaults on filespec.
Allow task to dynamically map memory-resident shared regions at run time. One address window of a task 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 (M,H)	SUPLIB=filespec/[-]SV[:apr]. Standard defaults on filespec.
Device	
Declare number of I/O (logical) units (M,H)	UNITS=decimal integer 0–250 UNITS=6

[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 one exists. This chart is intended only as a reminder of the options, their formats, and their defaults. Consult the *RSX–11M–PLUS and Micro/RSX Task Builder Manual* for full information on these options.

Table 6–1 (Cont.): Task Builder Options

Description and Language Key [1]	Format and Defaults [2]
Declare logical device assignment to units (M,H)	ASG=ddnn:unitnum: unitnum . . . ASG=SY0:1:2:3:4,TI0:5,CL0:6
Alterations	
Declare definition of global symbol (M)	GBLDEF:symbolname: symbolvalue. No default.
Declare global symbol reference (M)	GBLREF=symbolname. No default.
Declare series of patch values; patches I-space in I- and D-space tasks (M)	ABSPAT=segname:address: val1 . . . (up to 8). No default.
Declares series of patch values; patches D-space of I- and D -space tasks (M)	DSPPAT=segname:address:val1 . . . (up to 8). No default.
Declare series of patch values relative to global symbol (M)	GBLPAT=segname:symname[+/-offset]:val1 . . . (up to 8). No default.
Exclude the specified symbols from the symbol definition file of a supervisor-mode library (M)	GBLXCL=sym1:sym2: . . . :symn. No default.
Declare address and size of ODT SST vector(M)	ODTV= symbolname:vectorlength. No default.
Declare size of task SST vector (M)	TSKV=symbolname:vectorlength. No default.
Include specified symbols in STB file (M,H)	GLBINC=symbolname, symbolname, . . . ,symbolname. No default.

[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 one exists. This chart is intended only as a reminder of the options, their formats, and their defaults. Consult the *RSX-11M-PLUS and Micro/RSX Task Builder Manual* for full information on these options.

Examples

```
$ LINK RET
File(s)? WRAY RET
$
```

In this example, the object file WRAY.OBJ is built to produce 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/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 unit numbers (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/NOMEMORY_MAN WRAY [RET]
Option?PAR=KROBAR:50000:40000 [RET]
Option?[RET]
```

This example builds the task to run on an unmapped system in the partition named KROBAR, which has a base address of 50000 and a size of 40000.

```
$ LINK/CHECKPOINT:SYSTEM WRAY [RET]
$ LINK/CHECKPOINT WRAY [RET]
```

The commands in this example are equivalent. The task is built checkpointable to the system checkpoint file.

```
$ LINK/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.

Notes

You must read the appropriate documentation to use this command fully. This is true without exception.

The *RSX-11M-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.

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, any type of information.

Maintenance functions of LIBRARY include extracting, deleting, inserting, and replacing library entries. Libraries can be compressed 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 Relocatable Assembler, either directly or through MACRO. The assembler searches the library that you specify for macros listed in .MCALL statements and calls in your source program. It then searches the system macro library. See the *RSX–11M–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 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 sucessfully, you must familiarize yourself with this utility, which is described in the *RSX-11M-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.

For more information on how the MACRO-11 Relocatable Assembler and the Task Builder use libraries, see the *PDP-11 MACRO-11 Language Reference Manual,* the *RSX-11M-PLUS and Micro/RSX Task Builder Manual,* and the *RSX-11M-PLUS Guide to Program Development.*

You cannot use wildcard characters [the percent sign (%) and the asterisk (*)] in file specifications in LIBRARY operations.

Library modules are limited to 65,536 words per module.

Libraries are limited to 65,536 blocks.

The LIBRARY command operates on three kinds of libraries:

- Object module libraries. These can be user libraries or one of the following DIGITAL-supplied libraries:

LB0:[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

- 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

- 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.

LIBRARY

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 and vectored 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 that 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

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.

LIBRARY/COMPRESS

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.

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, a 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 library is being created.

infilespec[,s]

Specifies the file or files to be used as input to the new library file. If no input files are specified, an empty library file is created. The default file types are OBJ when creating object module libraries, MAC when creating macro libraries, and UNI when creating universal libraries.

Arguments

If you are specifying more than one argument, the arguments must be enclosed in parentheses and separated by commas. If you are specifying only a single argument, the parentheses are not necessary.

GLOBALS:n

Specifies the number of global symbols (entry point table entries) to allocate. The default is 512 for object libraries. This value is always forced to 0 for macro and universal libraries. The value of n can range from 0 to 4096.

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 to 4096.

LIBRARY/CREATE

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 included in the Task Builder's symbol table only 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.

The /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.

Example

```
$ LIBRARY [RET]
Operation? CREATE: (GLOBALS:128,MODULES:64) [RET]
Library? CARNEGIE [RET]
Modules? DEWEY,CONGRESS,MODERN [RET]
```

This example creates the library CARNEGIE.OLB, with a size of 100 blocks (the default), 128 entry points, and 64 module names, and inserts object modules from the input files DEWEY.OBJ, CONGRESS.OBJ, and MODERN.OBJ.

Notes

This command is the equivalent of applying the LBR /CR switch on the output file.

Tables should be allocated to maximum anticipated size. Expanding table allocations requires using LIBRARY/COMPRESS to copy the entire file.

If you use the qualifiers /SELECTIVE_SEARCH and /SQUEEZE 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 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, after you have used the /SQUEEZE qualifier, the code emerges as follows:

```
MOVB    #'
```

To preserve the meaningful semicolon, use the following form:

```
MOVB    #';,R1 ;THIS EXTRA SEMICOLON SAVES THE CODE
```

After you have used the /SQUEEZE qualifier, the code emerges as follows:

```
MOVB    #';,R1
```

Similarly, the /SQUEEZE qualifier causes the following comment to disappear:

```
;NEXT 5 LINES ARE CRUCIAL
```

The second semicolon preserves the text in the following line:

```
;NEXT 5 LINES ARE CRUCIAL;
```

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]

Parameters

libspec

Specifies the name of the library from which modules are to be deleted.

module[,s]

Specifies the modules that are to 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).

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.

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 with TI: to have the output printed on your terminal.

Examples

```
$ LIBR/EXTRACT/OUTPUT:MATH LB0:[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.

LIBRARY/EXTRACT

```
$ LIBR/EXTRACT/OUT:TI: LB0:[001001]RSXMAC.SML STOP$S RET
 .MACRO  STOP$S  ERR
 .MCALL  DIR$
 MOV     (PC)+,-(SP)
 .BYTE   131.,1
 DIR$    ,ERR
 .ENDM   STOP$S
```

This example extracts the macro named STOP$S from the system macro library and spools it to the terminal.

Notes

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

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, 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.

Macros that have been squeezed not only take up less room in the macro library file but also take up less memory.

LIBRARY/INSERT

See the Notes to Section 6.6.2 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.

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.

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-87 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words
```

```
ALERR
ALSCT
ALTPRI
ANSPAD
ARITH
    .
    .
    .
```

This example shows the default display from LIBRARY/LIST. Only the module names appear. The information appears on your terminal.

```
$ LIBRARY/LIST/FULL LBO:[001001]SYSLIB RET

Directory of file SYSLIB.OLB;1002
Object module library created by:  LBR V06.00
Last insert occurred 25-MAY-87 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-87  Ident:00

ALSCT   Size:00134  Inserted:22-MAY-87  Ident:00

ALTPRI  Size:00081  Inserted:22-MAY-87  Ident:00

ANSPAD  Size:00068  Inserted:25-MAY-87  Ident:20.0P6

ARITH   Size:00088  Inserted:22-MAY-87  Ident:03.03
    .
    .
    .
```

This example shows the display from LIBRARY/LIST/FULL, which includes full module descriptions. This information appears on your terminal.

```
$ LIBRARY/LIST/NAMES LBO:[001001]SYSLIB RET

Directory of file SYSLIB.OLB;1002
Object module library created by: LBR V06.00
Last insert occurred 25-MAY-87 at 19:34:32
MNT entries allocated: 768; Available: 549
EPT entries allocated: 2048; Available: 873
File space available: 00309 words
Recoverable deleted space: 14892 words

** Module:ALERR

  $ALERR

** Module:ALSCT

  ALSCT

** Module:ALTPRI

  ALTPRI

** Module:ANSPAD

  ..ANSP

** Module:ARITH
```

```
  $DIV    $MUL
    .
    .
    .
```

This example shows the display from LIBRARY/LIST/NAMES, which lists entry points for each module.

```
$ LIBRARY/LIST/FULL/NAMES LB0:[001001]SYSLIB RET

Directory of file SYSLIB.OLB;1002
Object module library created by:  LBR V06.00
Last insert occurred 25-MAY-87 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-87  Ident:00
  $ALERR
** Module:ALSCT   Size:00134  Inserted:22-MAY-87  Ident:00
  ALSCT
** Module:ALTPRI  Size:00081  Inserted:22-MAY-87  Ident:00
  ALTPRI
** Module:ANSPAD  Size:00068  Inserted:25-MAY-87  Ident:20:OP6
  ..ANSP
** Module:ARITH   Size:00088  Inserted:22-MAY-87  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/LIST

```
$ 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.

Notes

This command is the equivalent of applying the LBR /LI, /FU, or /LE switches to the listing file.

Note that there are four possible listing formats with this command. (See the command qualifiers at the beginning of this section.)

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 global symbols TINKER, EVERS, and CHANCE from the library DOUBLE.OLB.

Notes

This command is the equivalent of the LBR /DG switch applied to the output file.

If a specified global is not included in the library, the following message is printed on your terminal:

```
LBR -- *FATAL* -- No entry point named "name"
```

All the global symbols (entry points) in your list up to the incorrect name will have been deleted. None of the global symbols following the incorrect name will have been deleted. You should use the LIBRARY/LIST qualifier to determine which global symbols 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.

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 included in the Task Builder's symbol table only 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.

6.6.9 Error Messages from the LIBRARY Command

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 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 recreated. 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 either is not a library or is corrupted.

User Action: Make sure the file is a library. If the file is a library, use VFY, the File Structure Verification Utility, to determine if it is corrupted. See the *RSX-11M-PLUS Utilities Manual.*

LBR—Command I/O error

Explanation: There are two possible causes:

- The problem may be with the physical device, which, for example, may not be spun up.
- The file may be corrupted or have an incorrect format; for example, a record length may exceed 132_{10} 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.

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 entry-point or module-name 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 GLOBALS or MODULES argument to increase table size.

LBR—EPT or MNT space exceeded in compress

Explanation: An entry-point or module-name 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 name.

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 name.

LBR—Get time failed

Explanation: This is a 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: The command included an argument for GLOBALS or MODULES greater than 4096.

User Action: Check for proper syntax and reenter the command.

LBR—Invalid module format, insertion-module

Explanation: The command attempted to insert the named macro definition into an object module library.

User Action: Correct the error by reentering the command with an object module for an object module library or a macro for a 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 the command.

LBR—Invalid library type specified

Explanation: Your command attempted to set an invalid library type as the default. The file types OBJ and MAC are the only valid default file types for the command.

User Action: Reenter the command after checking the syntax.

LBR—I/O error on input file "filename"

Explanation: A read error occurred on an input file. This error can have two causes:

- A problem with the physical device; for example, it may not be spun up.
- The input file is corrupted or in the wrong format; for example, a record may exceed 132_{10} 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 deletion. In this case, the marking failed. The work file is a lost file and does not appear in any directory.

User Action: See your operator or system manager. The file must be deleted using VFY, the File Structure Verification Utility. See the *RSX-11M-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 the command for proper syntax and reenter it. You may have misspelled the entry point name or named the wrong library.

LBR—No module named "module"

Explanation: Your command named a module to be deleted that is not in the library.

User Action: Check the command for proper syntax and reenter it. You may have misspelled the module name or named the wrong library.

LBR—Open failure on file "filename"

Explanation: The file system detected an error while attempting to open a file. This error is caused by one of the following conditions:

- The user directory area is protected against a write operation.
- There is a problem with the physical device, which, for example, may not be spun up.
- The volume is not mounted.
- The directory does not exist.
- The file does not exist.
- There is insufficient space to allocate the library file on a LIBRARY/CREATE or LIBRARY /COMPRESS operation.
- 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 conditions can cause this error:

- The volume is full.
- The device on which the volume is mounted is write-protected.
- There is a problem with the physical device.
- There is insufficient pool space (DSR) in the Executive.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

LBR—Output error on filename

Explanation: A write error has occurred on the output file. One of the following three conditions can cause this error:

- The volume is full.
- The device on which the volume is mounted is write-protected.
- 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 conditions can cause this error:

- The volume is full.
- The device on which the volume is mounted is write-protected.
- 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.

Chapter 7
Running Tasks

The task is the fundamental executable programming unit. On RSX–11M–PLUS operating systems, the term *task* refers to both tasks built from user programs and system tasks, such as compilers or utilities.

Tasks are built (linked) from one or more object modules. 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).

Users can install their own tasks with the RUN command. These tasks remain installed as long as they are executing and are removed from the system when they finish.

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, 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
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 (see 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 to 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 a maximum of 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 (removed from memory and sent 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 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.

Consult the *RSX-11M-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 to 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.

The following command, passed through DECnet by host terminal number 6:

```
SHOW TIME
```

results in a task name SHOR6 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, 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]
```

gives your task a name in the form PIPT10.

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 to 77 in octal numbers. The system supports as many as 256_{10} terminals. Terminals 65 to 256 are numbered 100 to 377_8. 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 invalid task name. Therefore, when the RUN command is issued from terminals numbered 100 to 377_8, 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
A0–A7	100–107
B0–B7	110–117
C0–C7	120–127
D0–D7	130–137
E0–E7	140–147
F0–F7	150–157
G0–G7	160–167

Task Number	Terminal Unit Number
H0–H7	170–177
I0–I7	200–207
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:

- 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.
- To install, run, and remove after execution a task from a task image file stored in the system directory or library directory.
- To run immediately a task previously installed by a privileged user.
- 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 in to memory and execution commences. When the task has executed, the image is removed from memory and from the STD.

Whenever a RUN command includes any element of a file specification, this procedure is followed. The following command:

```
$ RUN DU0:MUPPET [RET]
```

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. 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, unvectored 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 TKBFSL, TKBRES, or TKB.
2. Install and run the task image.
3. Remove the task image when it has completed execution.

Note that the system searches for file names FSL and RES before TKB. For each type of task, it searches the library directory first and then the system directory.

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.

7.2.1 Running Tasks Contained in Task Image Files

When used to run an uninstalled task, RUN performs the functions of the INSTALL and REMOVE commands in addition to its own. This command is the only one a nonprivileged user can use to 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 ($) directs the system to search first for the file in the library directory and then in the system directory. In this case, the dollar sign ($) specifies a device and directory and thus counts as an element of a file specification.

Command Qualifiers

These qualifiers establish how the uninstalled task is to be installed.

/[NO]CHECKPOINT

Specifies whether or not the task is to be run checkpointable. The default is established when the task is built (linked). This qualifier is used to override the default, allowing you to run checkpointable tasks as noncheckpointable, or vice versa.

/COMMAND:"taskcommand"

Passes the specified command line to the task you are running. The command must begin with the task's acronym and it must be valid for the task. Otherwise, a syntax error message will be returned. The command is also limited to 40 characters in length and must be inside the quotation marks ("). For example, the following command runs PIP and then passes the /LI switch to it:

```
$ RUN/COMMAND:"PIP /LI" $PIP RET
```

When the task has finished executing the command, the task is removed.

/EXTENSION:n

Specifies that n additional decimal words of address space be allocated to a task.

/[NO]IO_PAGE

Specifies that a privileged task can overmap the I/O page. If you specify /IO_PAGE, RUN assumes that the task will overmap the I/O page and if the task is larger than 8K words issues a warning message. If you specify /NOIO_PAGE, RUN assumes that the task does not need to use the I/O page; no warning message is issued. The default is /IO_PAGE.

/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 when the task is built 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-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_{10}. 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 to 250_{10}.

RUN

/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.

/[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 when the task is built.

/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 is useful in batch jobs or in indirect command files that run installed tasks, where the command 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 uppercase letters A to Z, the numerals 0 to 9, 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 the protection class in which the task belongs and thus determines file access. The 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, in which the file type is 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 TASKS/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_8, in this case, TT107:. Notice the name given to the SHOW task.

In the following example, the terminal number is less than 77_8.

```
$ 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:BINDER ASIMOV [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.

Notes

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, because 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 CO0: as their TI:. For this reason, these commands are privileged. Such tasks send output to CO0: and not to the terminal from which they were run.

Format

```
RUN[/qualifier[s]]
Task?  taskname

RUN[/qualifier[s]] taskname
```

Command Qualifiers

```
/DELAY:nu
/INTERVAL:nu
/SCHEDULE:hh:mm:ss
/STATUS:arg
        COMMAND
        TASK
/SYNCHRONIZE:u
/UIC:[g,m]
```

Parameter

taskname

Specifies the name of the installed task to be run. The names of installed tasks are one to 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.

/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 is useful in user batch jobs or in indirect command files that run installed tasks, where the command 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 specified only 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 the file protection class in which the task belongs 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.

```
$ 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 batch job. The job installs the task HIFI from the library directory and then processes it. 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 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.

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 rebootstrapped.

RUN

Error Message

RUN—Invalid time parameter

Explanation: The command specified an invalid time argument.

User Action: Check for proper syntax and reenter the command.

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.

/[NO]POSTMORTEM

Specifies that a postmortem dump of the task be taken (or not taken) before it is aborted. See the *RSX-11M-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 INFORM THE KLINGON 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 issued the ABORT command at the DCL prompt. 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.

```
$ DIRECTORY *.RNO [RET]

Directory DU2:[303,5]
20-MAY-87 16:28

ALTCLI.RNO;3          213.       27-APR-87 16:52
ANNI.RNO;6            3.         30-MAR-87 09:30
APRCOM.RNO;4          9.         15-APR-87 10:35
BEST.RNO;5            40.        28-APR-87 15:50
CATCH.RNO;4           7.         22-APR-87 09:39
CHAP11SMG.RNO;13      49.        13-MAY-87 16:36
CHAP12SMG.RNO;51      223.       08-APR-87 13:15
CHARSET.RNO;5         6.         15-MAY-87 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 searched for was present. Rather than wait for the directory listing to complete, the user typed CTRL/C and entered the ABORT command in response to the 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/POSTMORTEM RUN [RET]
14:41:11 Task "TT10" terminated
         Aborted via directive or CLI
         R0=000000
         R1=100077
         R2=135600
         R3=000000
         R4=000000
         R5=000000
         SP=012540
         PS=170017
$
```

In this example, the user requested a postmortem dump. The contents of the task's registers at the time of the abort are listed on the terminal along with notification of the dump. The dump file will be printed on the system's line printer.

Notes

A is the short form of ABORT. See also the discussion of CTRL/C in Chapter 3, under SET TERMINAL.

ABORT

ABORT forces an orderly termination of a task. To effect the termination, the system performs four steps:

1. Alters the task's priority to 247. A task must be memory resident to be aborted. This step usually forces the quick return of a checkpointed task.
2. Performs I/O rundown: cancels I/O for non-file-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 several 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.

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 an SPND$S directive within the task. A RSUM$ 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
   REGS 0-6 001372 000037 000012 140311 001254
$ CONTINUE [RET]
"Message from Yoyo:  Not suspended. Climbing string"
$
```

In this example, the user ran a task that issues a message and then suspends itself. Not all suspended tasks issue such messages, but you can identify a suspended task through SHOW TASKS/ACTIVE. The user issued SHOW TASKS/ACTIVE and named the task. Included in the listing under STATUS on the second line, the mnemonic SPN verifies that the task was suspended. The user issued CONTINUE without naming the task. Because the task was initiated by RUN, CONTINUE defaulted to a task name based on the name of the terminal.

CONTINUE

Notes

A task cannot suspend any task other than itself.

For more information on task states, see the *RSX-11M-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-PLUS Batch and Queue Operations Manual* for more information.

Error Message

CON—Task not suspended

Explanation: The command attempted to continue an unsuspended task.

User Action: None.

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 cancel only 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, 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 [RET]
ARDVRK Scheduled at 17-MAY-87 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 5 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 command would have had no effect.

Note

See Section 7.14.5 for information on SHOW CLOCK_QUEUE.

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...
SHOT5
TT5
$ STOP/BLOCK [RET]
$ SHOW TASK:TT5/FULL [RET]
TT5     100604   GEN     073464 00541000-005436 PRI - 50.  DPRI - 50.
   STATUS:  BLK -CHK WFR -PMD REM MCR
   TI - TT5:  IOC - 0.  BIO - 0.  EFLG - 000001 000000 PS - 170000
   PC - 002206 REGS 0-6 001637 040066 040060 140311 163500 000000 001252
$
$

$ START/UNBLOCK [RET]
$ SHOW TASK:TT5/FULL [RET]
TT5     100604   GEN     073464 00541000-005436 PRI - 50. DPRI - 50.
   STATUS:  -CHK WFR -PMD REM MCR
   TI - TT5:  IOC - 1.  BIO - 0.  EFLG - 000001 000000 PS - 170000
   PC - 002206 REGS 0-6 006173 042066 040060 140311 163500 000000 001260
$
```

In this example, the user on terminal TT5: ran a task with the RUN command, giving the task the name TT5 by default. Then the user issued the STOP/BLOCK command, without naming the task. The /FULL 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-PLUS and Micro/RSX System Management Guide* for more information.

7.7 START

The START command resumes execution of a task that was stopped by the Stop (STOP$S) 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 task name, 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.

Note

Several other forms of the START command are used to control the Queue Manager. See the *RSX-11M-PLUS and Micro/RSX System Management Guide* for more information. For information on STOP$S, see the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual.*

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 command unblocks the task initiated with the RUN command and named after the terminal if this task is 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 unblock tasks blocked by the Executive while waiting for system resources or for synchronization. You can only unblock tasks blocked with the STOP/BLOCK command.

The START/UNBLOCK and ABORT commands are the only means of canceling the STOP /BLOCK command.

The system automatically unblocks a blocked task when the blocked task is aborted.

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 by means of the RUN command or a variety of Executive directives, including Request Task (RQST$), Run Task (RUN$), and Spawn (SPWN$).

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
/[NO]DEFER_BINDING
/EXTENSION:n
/FAST_MAP
/[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 library directory and in the system directory. If you do not include the /TASK_NAME qualifier, the task will be installed under a name based on the first six characters of the file name unless another name was assigned through the TASK= option of the Task Builder. See the examples.

Command Qualifiers

/[NO]CHECKPOINT

Specifies whether or not the task is to be checkpointable. The default is set at link time. This qualifier overrides the link-time checkpointability specification.

/[NO]DEFER_BINDING

Specifies whether or not the task's header is bound to Executive data structures when the task is installed. The /DEFER_BINDING qualifier, which is the default, specifies that binding to the Executive will be deferred until the task is loaded into memory. Use of this qualifier resolves problems that occur when a task is installed in more than one system (running or disk resident) at the same time. The /NODEFER_BINDING qualifier specifies that binding occurs at the time the task is installed.

/EXTENSION:n

Specifies that n (decimal) additional words of address space are to be allocated to a task.

/FAST_MAP

Specifies that space is allocated in memory between the task and the external header for use by the fast-mapping feature of the Executive.

/[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 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-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 to 250_{10}.

/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 sent to a slaved task is marked with the TI: of the sending task. When the installed task receives the data, the system sets the task's TI: to that associated with the data. The default is set at link time. This qualifier overrides the link-time specification.

Note that data may not be sent to a slaved task running on a virtual terminal.

/TASK_NAME:*taskname*

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

/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 DEC Multinational Character Set. This qualifier is privileged.

INSTALL/TRANSLATION_ROUTINE attaches ancillary control drivers (ACDs) 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 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 United States keyboard
MCSVT.TSK	For VT100 non-W model terminals that have the DEC 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.

/[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
```

In response to the command in this example, the system 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
```

The command in this example directs the system to look in the library directory (by convention, [3,54]) and in the system directory (by convention, [1,54]) 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
```

The command in this example directs the system to look 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
```

The command in this example directs the system to look 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]
```

The command in this example directs the system to install the task image file named KILROY.TSK in a partition named LEROY.

```
$ INSTALL/CHECKPOINT/POSTMORTEM/PRIORITY:75/TASK:LARGO KEY [RET]
```

The command in this example directs the system to look 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_{10}. If the task should exit unexpectedly, a postmortem dump is generated.

Notes

REMOVE counteracts INSTALL. See Section 7.9.

Tasks can also be installed through the install-run-remove form of the RUN command. Such tasks remain installed only while they are executing. While such tasks are installed, they are similar to other installed tasks. You cannot, however, count on the presence of a task installed through install-run-remove.

Several situations require tasks to be explicitly installed through INSTALL. If you 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-PLUS and Micro/RSX Executive Reference Manual.*

Error Messages

INS—Access to common block denied

Explanation: The 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: The command tried to install an ACD that is currently installed. The Ancillary Control Driver task (ACD) returns the message.

User Action: None.

INS—Addressing extensions not supported

Explanation: The command tried to install a task with a 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-word 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.

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.

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: The 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: The command attempted to install a task from a noncontiguous file. Task images must be contained in contiguous files.

User Action: Ensure you have specified the correct file and reenter the command.

INS—File not task image

Explanation: The command attempted to install a task from a file whose label block indicated it was not a task image.

User Action: Ensure you have specified the correct file and reenter the command.

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: The command specified a device that was not a valid task-residence device.

User Action: Ensure that you specified the proper device and 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 words; using APR 5, the task was larger than 8K words. If the Executive has been built to support 20K words 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-PLUS and Micro/RSX Task Builder Manual* for a description of the /PR switch (/PRIVILEGED qualifier to LINK).

INS—Illegal priority

Explanation: The command specified a priority value of less than 0 or greater than 250.

User Action: Reenter the command after checking for proper syntax.

INS—Illegal use of parname partition or region

Explanation: The command attempted to install a task into a CPU or secondary pool partition.

User Action: Reenter the command, specifying the 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.

INS—No checkpoint space, assuming not checkpointable

Explanation: This is a warning message on systems that do not support dynamic allocation of checkpoint space. The 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 (pool space) was available for the Common Block Directory (CBD) or the 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 (pool space) was available for a new Task Control Block (TCB) in the System Task Directory (STD). The 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_8 to map the 16K-word Executive or virtual 120000_8 to map the 20K-word 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-PLUS and Micro/RSX Task Builder Manual* for a description of the /PR switch (/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 the task's partition.

User Action: If you want to, you can create the missing partition.

INS—Specified partition too small

Explanation: The command attempted to install a task in a partition that was smaller than the task.

User Action: 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 space, 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.

INSTALL

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 corrupted 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: The 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.

INS—Too many LUNS

Explanation: The task requested more than 250 logical unit numbers (LUNs) to be assigned.

User Action: Recheck the task build and code for the 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.

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 Qualifiers

/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 Qualifiers

/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

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 detaches all attached regions. See the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for more information.

Error Messages

ACD—ACD is busy but was marked for delete

Explanation: The 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.

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
$
```

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.

FIX

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 reentrant.

Overlaid tasks may be fixed and, if their root segments are serially reentrant, 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.

7.11 UNFIX

UNFIX frees a fixed task or region from memory. This allows tasks that 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.

Note

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.

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:.

Note

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: The 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.

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 issued the Executive directive Specify SST Vector Table for Debugging Aid (SVDB$).

DIGITAL offers two debuggers for use with this command:

- ODT-11, the On-Line Debugging Tool
- RSX–11M–PLUS FORTRAN–77 DEBUG

See the *RSX–11M–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 from which the task was run.

Format

DEBUG [taskname]

Parameter

taskname

Specifies the name of the task you want to trap. If you do not specify a task name, 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.

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-PLUS and Micro/RSX Executive Reference Manual* for information on the SVDB$ directive.

See the *RSX-11M-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.

SET PARTITION

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.

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 Command Qualifiers

/BASE:n
/SIZE:n

Command Qualifier

/TOP:arg

Partition Types

/DEVICE
/DIAGNOSTIC
/SYSTEM

Required Parameter

parname

Partition names are from one to six Radix–50 characters. Each partition (or subpartition) must have a name. SET NOPARTITION eliminates the named partition.

Required Command 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.

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.

For system-controlled partitions, the values in Table 7–1 can range from 0 to 1920.K, or any valid representation of these limits. The upper limit is the size of your system's memory.

Table 7–1: Base and Size Specifications for SET PARTITION

Format			Calculated Value	Base 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

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.

/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.

SET PARTITION

/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_8 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–PLUS systems and making the best use of available memory.

You can specify the partition in which you want your task to run 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 (Chapter 8).

See the *RSX–11M–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: Check the command for proper values and reenter it.

SET—Nonexistent memory

Explanation: The command attempted to define a partition in nonexistent memory.

User Action: Check the command for proper values and reenter it.

SET—Partition already exists

Explanation: The command attempted to define a partition with a name already in use.

User Action: Use another name.

SET—Space used

Explanation: The command attempted to create a partition in a storage area already occupied.

User Action: Check the command for proper values and reenter it.

SET—Task installed in partitions

Explanation: The command attempted to eliminate a partition containing an installed task.

User Action: Determine the cause of the error and proceed accordingly.

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
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 in secondary pool

contentname

Indicates one of the following:

[taskname]	An inactive resident task
<TASKNAME>	An active task
(dd:)	A specific device driver
+xxxx+	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  >
                          .
                          .
                          .
```

This example shows the display from SHOW PARTITIONS when no partition is named. Information on all partitions and dynamic task subpartitions in the system is displayed. DRVPAR contains a number of device drivers, each in its own subpartition. Notice that SYSPAR's base address is 00201000 and its size is 10000_8 bytes. See the next example.

SHOW PARTITIONS

```
$ SHOW PARTITION: SYSPAR [RET]
PAR=SYSPAR:002010:000100:SYS
```

This example shows the display from SHOW PARTITIONS when a partition is named. No information about subpartitions or about any tasks in the partition is displayed. Notice that SYSPAR's base address is given as 002010 and its size is 100 64-byte blocks.

Notes

SHOW PARTITIONS unadorned is the equivalent of the MCR command PAR.

SHOW PARTITIONS with a partition name is the equivalent of the MCR SET /PAR=parname command.

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 information about tasks active on your terminal can be displayed in full or brief format. Section 7.14.3.2 explains how information about installed tasks can be displayed in full or brief format. Section 7.14.3.3 explains how information about tasks installed from a specified device can be displayed. Sections 7.14.3.4 and 7.14.3.5 show how dynamic (continously updated) information about active tasks can be displayed. Section 7.14.3.6 explains the display of the logical unit numbers (LUNs) of an installed task.

SHOW TASKS/ACTIVE

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.

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:)
DBOFCP   (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 TASKS/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 r1val r2val r3val r4val r5val r6val
```

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 the 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 tasks with the SHOW TASKS/ACTIVE /DYNAMIC and SHOW TASKS:taskname/DYNAMIC commands (see Sections 7.14.3.4 and 7.14.3.5).

Table 7–2: Task Status Flags

Status Flag	TCB Flag	Meaning
ABO	T2.ABO	Being aborted
ACP	T3.ACP	Ancillary Control Processor
AST	T2.AST	AST state
BLK	TS.STP	Blocked externally by CLI command
CAF	T2.CAF	Dynamic checkpointing allocation failure
CAL	T3.CAP	Checkpoint space allocated in task image
-CHK	T2.CHK	Not checkpointable
CIP	TS.CIP	Blocked for checkpoint in progress
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

Table 7-2 (Cont.): Task Status Flags

Status Flag	TCB Flag	Meaning
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
NRP	TS.NRP	Mapped to nonresident partition
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.PRV	Privileged
PRV	T4.PRV	Task was privileged but has cleared T3.PRV
RDN	TS.RDN	I/O being run down
REM	T3.REM	Remove on exit
REX	T2.REX	Abort AST effected or in progress
ROV	T3.ROV	Resident overlays
RST	T3.RST	Restricted—used by layered software
RUN	TS.RUN	Running on another processor
SEF	T2.SEF	Stopped for event flag
SLV	T3.SLV	Slaved

Table 7-2 (Cont.): Task Status Flags

Status Flag	TCB Flag	Meaning
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

Note

These status flags are displayed by several forms of the SHOW TASKS 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 sign (-) 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-PLUS and Micro/RSX Task Builder Manual*, the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual*, and the *RSX-11M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual* for more information. The CDA manual has a full explanation of the contents of the TCB.

Examples

```
$ SHOW TAS/ACT/FULL TT7 [RET]
TT7  066734  GEN  064244  01676500-02062100  PRI - 50.  DPRI - 50.
   STATUS:  -CHK WFR -PMD REM MCR
   TI - TT7: IOC - 1. EFLG - 000000 040000  PS-70004  PC-072132
   REGS 0-6  041130 051172 000000 000040 010400 051520 001472
```

This example shows the display in full format for the task named TT7 (initiated by a RUN command from terminal TT7:). The task is running in the default partition GEN at a priority of 50. The task is not checkpointable. It is in a "wait-for" state. No postmortem dump 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/FULL [RET]
...LDR 034210  LDRPAR 033774 00364000-00367000  PRI - 248.  DPRI - 248.
   STATUS:  STP - PMD PRV NSD FXD
   TI - C00:  IOC - 0.  EFLG - 000001 000000  PS - 170000  PC - 120526
   REGS 0-6  000162 024207 177777 066640 066610 066314 120250
RMDEM0 046320  GEN    046640 01367700-01415100  PRI - 225.  DPRI - 225.
   STATUS:  -CHK WFR -PMD PRV MCR
   TI - TT62:  IOC - 0.  EFLG - 000035 040000  PS - 170010  PC - 125014
   REGS 0-6  000002 131316 131436 000000 135055 000000 121240
MCR... 116464  SYSPAR 036330 00201000-00211000  PRI - 160.  DPRI - 160.
   STATUS:  STP -PMD PRV CLI NSD CAL
   TI - TT0:  IOC - 0.  EFLG - 000001 040000  PS - 170000  PC - 124412
   REGS 0-6  000000 120556 000400 120512 000100 120572 120442
DCL... 115530  GEN    037344 00624700-00634700  PRI - 160.  DPRI - 160.
   STATUS:  STP -PMD PRV CLI NSD CAL
   TI - TT10:  IOC - 0.  EFLG - 100001 040000  PS - 170000  PC - 124470
   REGS 0-6 025550 025550 000000 120426 123530 000000 120362
SHOT10  066314  GEN    066610 02221700-02261700  PRI - 160.  DPRI - 160.
   STATUS:  -PMD REM PRV
   TI - TT10:  IOC - 0.  EFLG - 000001 040000  PS - 170017  PC - 121202
   REGS 0-6  000000 131574 050712 000000 000000 000000 001216
DB00F1 117570  GEN    036770 00537000-00624700  PRI - 149.  DPRI - 149.
   STATUS:  STP ACP -PMD PRV NDS CAL ROV
   TI - C00:  IOC - 0.  EFLG - 000002 040001  PS - 170000  PC - 120546
   .
   .
   .
LPP0   061444  GEN    062504 00725200-00740000  PRI - 80.  DPRI - 80.
   STATUS:  STP PRV
   TI - C00:  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 - C00:  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 - C00:  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 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.

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 is 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.

/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 LBO:-00163310 FIXED
TKTN     04.7     TKNPAR  248. 00010000 LBO:-00171022
RMDEMO   V1.03    GEN     200. 00026700 LBO:-00171022
MTAACP   0013     GEN     200. 00013600 EMO:-00000315
F11MSG   V0010    GEN     200. 00025500 LBO:-00167207
NETACP   V02.00   GEN     200. 00025500 DBO:-00644640
EVP...   V01.00   GEN     199. 00013100 DBO:-00644640 CHECKPOINTED
...MCR   1.1      GEN     160. 00020300 LBO:-00162467
SHOT5    1.1      GEN     160. 00020300 LBO:-00162467
...MOU   2502     GEN     160. 00037700 LBO:-00222416
...DCL   00       GEN     160. 00035600 DBO:-00212612
MCR...   2.02     SYSPAR  160. 00010000 LBO:-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 the 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 REGS 0-6 000000 120476 000102 120516 057020 040220 120366
```

This example shows the full format of the display from SHOW TASKS/INSTALLED. In this case, the user named a single task. The default is to show the full display for all installed tasks. The status flags in the second line of the display are explained in Table 7–2.

Note

The brief format of SHOW TASKS/INSTALLED is the MCR TAS command. The full format of SHOW TASKS/INSTALLED is the MCR TAL command.

SHOW TASKS/INSTALLED/DEVICE

7.14.3.3 SHOW TASKS/INSTALLED/DEVICE

SHOW TASKS/INSTALLED/DEVICE displays the names and status of all tasks installed from a specified device.

Format

SHOW TASKS/INSTALLED/DEVICE:ddnn:

Example

```
$ SHOW TASKS/INSTALLED/DEVICE:DR0: RET
DR00F1 M0323  SYSPAR 149. 00055700 LB0:-00475656 FIXED
F11ACP M0323  GEN    149. 00044000 LB0:-00315566
DB01F1 M0323  GEN    149. 00044000 LB0:-00315566
DB07F1 M0323  SYSPAR 149. 00044000 LB0:-00315566 FIXED
ERRLOG V1.01  GEN    148. 00042200 LB0:-00335276
PMD... 04.0   GEN    140. 00015200 LB0:-00503535
HRC... 02     GEN    140. 00055100 LB0:-00505331
SYSLOG 1.02   GEN    130. 00016300 LB0:-00504250
...INS 06     GEN    100. 00035400 LB0:-00502750
```

This display contains the following information from left to right:

- Task name
- Task version identification
- Partition name
- Task priority
- Task size
- Load device identification
- Logical block number on the load device
- Task memory state

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 task with higher priority.

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/ACTIVE/DYNAMIC

SHOW TASKS/ACTIVE/DYNAMIC displays continously updated information on the Active Task List on a video terminal or a static display on a hardcopy terminal.

This function is performed by a task called the Resource Monitoring Display (RMD). For more information on RMD, see Chapter 8 and the *RSX-11M-PLUS and Micro/RSX System Management Guide*. RMD has the following display pages:

- Task, which displays a task header
- Active, which displays all or part of the Active Task List
- Memory, which displays the contents of memory
- Help, which provides help on RMD

Once you have invoked any one of these pages, 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.

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, switch to the Memory page or the Active page, and then switch back to the Task page, where the named task will still be displayed.

This command is privileged if 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.

Format

SHOW TASKS/ACTIVE/DYNAMIC[/qualifier[s]]

Command Qualifiers

/OWNER:arg
        ddnn:
        ALL
/PRIORITY:n
/RATE:n

Command Qualifiers

/OWNER:arg
ddnn:
ALL

Specifies the terminal or other device from which the tasks displayed are being run. You can name any device, pseudo device, logical device name, or terminal number for this argument. The default argument is ALL, which displays tasks owned by all devices on the system.

You can change the owner displayed by pressing the ESC key and entering a new device name.

SHOW TASKS/ACTIVE/DYNAMIC

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 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.

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
```

The screen is cleared and the following display appears:

```
RSX-11M-PLUS      V4.0 BL40   512K                    14-MAY-87 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
SHOH0  00026700 HT0    225.   0. -CHK -PMD  PRV  MCR
NETACP 00025500 C00    200.   0.  STP  ACP -PMD  PRV  NSD
EVP... 00014200 C00    199.   0.  STP -PMD  PRV
DCLT5  00035600 TT5    160.   0. -CHK  STP  REM  PRV  CLI
DBOFCP 00044000 C00    149.   0. -CHK  STP  ACP -PMD  PRV  NSD
                       .
                       .
                       .
LP0    00014100 C00    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, its 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      V4.0 BL40   512K                    14-MAY-87 14:57:55

 Name  Length   TI     Pri    I/O Status flags
SHOT5  00026700 TT5           0.  CHK -PMD  REM  PRV  MCR
```

This example displays the Active Task List for a single terminal. The only task active at the terminal is the SHOW command itself.

Note

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.

7.14.3.5 SHOW TASKS:taskname/DYNAMIC

SHOW TASKS:taskname/DYNAMIC displays continuously updated information on a task's activity on a video terminal or a static display on a hardcopy terminal.

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.

Format

SHOW TASKS:taskname/DYNAMIC[/qualifier]

Command Qualifier

/RATE:n

Parameter

taskname

Specifies the task you want to inspect.

Command Qualifiers

/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          V4.0 BL40 512K                  14-MAY-87  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

R0 = 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:
```

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 on 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, because 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/LOGICAL_UNITS

SHOW TASKS:taskname/LOGICAL_UNITS displays the static logical unit number assignments (LUNs) for an installed task.

Format

SHOW TASKS:taskname/LOGICAL_UNITS

Parameter

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/ACTIVE/DYNAMIC command. (See Section 7.14.3.4.)

For more information on logical unit numbers, see Chapter 5. See also the *RSX-11M-PLUS and Micro/RSX Task Builder Manual.*

SHOW TASKS/LOGICAL_UNITS is the equivalent of the MCR LUN command.

Tasks can dynamically alter their LUN assignments through the Executive directive Assign LUN (ALUN$).

Error Messages

SHO—No LUNS

Explanation: This is an informational message. The task did not have any logical unit numbers assigned.

User Action: None.

SHO—Task not in system

Explanation: This is an informational message. The 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/ACTIVE/DYNAMIC can display LUN assignments for tasks installed with the install-run-remove form of RUN.

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 range from 1 to 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 TT7 [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
```

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 to indicate that it was a decimal number. 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_{10}, again omitting the decimal point. This time the command was processed without an error message, but the priority was changed to 255_8, which is 149_{10}.

Notes

SET PRIORITY changes both the running and default priority of a task. A task can change its own priority through the Executive directive Alter Priority (ALTP$). 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.

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-87 17:41:31:51 RESCHEDULE INTERVAL 05 SECS
ONEUP  SCHEDULED AT 29-MAY-87 17:48:44:05
CALDOR SCHEDULED AT 29-MAY-87 18:00:00:00 RESCHEDULE INTERVAL 02 HOURS
TEWKSY SCHEDULED AT 30-MAY-87 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.

7.14.6 SHOW COMMON

SHOW COMMON displays the names of resident commons installed in the system, their Partition Control Block (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]

Parameter

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 (CBD) 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

SHOW COMMON

Status Bit	Meaning
NSF	Not shuffleable
OUT	Out of memory
PER	Parity error
PIC	Position independent
RON	Read-only common

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.

Note

SHOW COMMON is the equivalent of the MCR CBD command.

7.15 SET and SHOW

You can set and show group global event flags. These are a set of 32 global event flags available only to tasks run from one group of UICs. Otherwise, these flags are identical to other global event flags. Event flags are used for communication between tasks. See the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for further information.

7.15.1 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.

Example

```
$ SHOW GROUPFLAGS RET
7          0        000000
200        0        000000
$
```

SET GROUPFLAGS

```
$ SET GROUPFLAGS RET
Flag? 303 RET
$
$ SHOW GROUPFLAGS 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.

7.15.2 SHOW GROUPFLAGS

SHOW GROUPFLAGS displays the group global event flags currently in the system.

Format

SHOW GROUPFLAGS

Example

```
$ SHOW GROUPFLAGS RET
7          1        000000 000000
200        1        000000 000000
201        1        000000 000000
303        1        000000 000000
333        1        000000 000000
```

This example shows the display from SHOW GROUPFLAGS. The first column is the group number with which the flags are associated. The second column is the access count. The two 6-digit numbers are the octal words that give the current state of each group global event flag.

Group global event flags are flags 65 through 96. The first word starts at flag 80 and goes right to flag 65. The second word starts at flag 96 and goes right to flag 81. The final column is reserved for the delete flag DEL, which means the group global event flags are marked for deletion and are not available. See the previous section.

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-PLUS MCR Operations Manual,* and in the *RSX-11M-PLUS and Micro/RSX System Management Guide.*

8.1 SET AND SHOW

You can set and display the time and date, the system and library directories, 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 mm 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 [RET]
Time? 8:05 [RET]
```

This example sets the time at 8:05 on the current day.

```
$ SET DAY [RET]
Date? 16-JUN-87 [RET]
```

This example sets the date at June 16, 1987. The time setting is unchanged.

```
$ SET DAY [RET]
Date? 06/14/87 17:14 [RET]
```

This example sets the date at June 14, 1987, and the time at 17:14:00. The date will not be displayed in the format entered but as 14-JUN-87.

```
$ SET DAYTIME 06-JUN-87 04:00:00 RET
$ SET TIME 04:00:00 06-JUN-87 RET
```

The examples are equivalent. Both set the date at June 6, 1987, and the time at 04:00 hours.

Note

Regardless of the prompt, you can enter the date or time in either order.

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-87
```

This example shows the date as June 6, 1987, and the time as 04:00:22.

8.1.3 SET SYSTEM

SET SYSTEM establishes certain characteristics of the system. It is a privileged command.

Format

SET SYSTEM[/qualifier]

Command Qualifiers

/[NO]CRASH_DEVICE:ddnn:/REGISTER=nnnnnn
/EXTENSION_LIMIT:n
/[NO]LOGINS
/NETWORK_UIC:[directory]
/PACKETS:n
/POOL:top
/POOL/LIMITS:arg
HIGH=n
LOW=n
MINIMUM_SIZE=n
TASK_PRIORITY=n
/DIRECTORY:[directory]

Command Qualifiers

/[NO]CRASH_DEVICE:ddnn:/REGISTER=nnnnnn

Loads a specific crash driver into a main memory partition and updates the crash database. To display the current crash dump device, omit the device specification ddnn:. The /NOCRASH_DEVICE qualifier unloads the current crash driver. You use the /REGISTER qualifier to specify the control and status register (CSR) of the desired device when the device is not currently on the system.

Note that this qualifier is valid only for pregenerated operating systems. For more information, see the *RSX-11M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual.*

/EXTENSION_LIMIT:n

Establishes the maximum size to which a task can extend itself by means of the Extend Task (EXTK$) 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-PLUS and Micro/RSX Task Builder Manual* and the *RSX-11M-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:[directory]

Specifies the directory in which all DECnet-related tasks are stored. This qualifier applies only to systems that selected the external communication products option during system generation. The brackets are required.

SET SYSTEM

/PACKETS:n

Specifies the maximum number of I/O packets to be retained in a separate queue for use by the Queue I/O Request (QIO$) directive. Changing the number of retained packets helps optimize the servicing of QIO requests. The argument n can range from 0 to 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.

SHOW SYSTEM/PACKETS displays the limit and also the number of packets currently available. See Section 8.1.4.

/POOL:top

Increases the size of the system pool (the dynamic storage region). The top argument is the number of 32-word blocks (decimal).

/POOL/LIMITS:arg[,s]13
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_{10} bytes.
LOW	The low pool limit in bytes. The default is 600_{10} 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_{10} 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_{10}.

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-PLUS and Micro/RSX 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.

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 (EXTK$) directive to 1200_8 32-word blocks (decimal) or 30K words.

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-PLUS MCR Operations Manual* and the *RSX-11M-PLUS and Micro/RSX 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 (EXTK$) system directive. The size is displayed as an octal number of 32-word blocks (decimal).

/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 Queue I/O Request (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, as follows:

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 follwing 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

/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

SET LIBRARY/DIRECTORY

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 library directory is searched first and then the system directory.

8.2.1 SET LIBRARY/DIRECTORY

SET LIBRARY/DIRECTORY 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 information on the relationship between SET LIBRARY /DIRECTORY and SET SYSTEM/DIRECTORY.

Format

SET LIBRARY/DIRECTORY:[directory]

8.2.2 SHOW LIBRARY

SHOW LIBRARY 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.

Format

SHOW LIBRARY

Example

```
$ SHOW LIBRARY RET
LIBUIC=[3,54]
```

This example shows the system library directory as [3,54].

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. The third column is the login UIC. HT2: is a DECnet host terminal.

```
$ SHOW USERS RET

TT7:    [7,40]      [7,40]      7-MAY-87 10:57    0     J. ANGLE
TT10:   [303,5]     [7,100]     7-MAY-87 11:10    2     W. WRITERS
TT14:   [7,25]      [7,25]      7-MAY-87 13:56    0     D. DUCK
TT15:   [1,1]       [7,50]      7-MAY-87 11:06    0     G. VOSKA
TT16:   [314,362]   [7,14]
TT17:   [UYAK]      [7,60]      7-MAY-87 11:15    0     P. STROM
TT20:   [12,10]     [7,24]      7-MAY-87 10:47    0     D. UYAK
TT21:   [1,6]       [7,13]      7-MAY-87 10:50    1     P. KROBAR
TT27:   [303,2]     [303,2]     7-MAY-87 12:55    0     C. FUTILE
TT30:   [FRINI]     [303,10]    7-MAY-87 12:33    0     J. FRINI
TT32:   [VEISS]     [7,11]      7-MAY-87 14:14    1     P. VEISS
TT33:   [7,16]      [7,16]      7-MAY-87 10:56    0     S. THOMS
TT34:   [1,24]      [7,21]      7-MAY-87 11:08    0     N. EPPS
TT35:   [KNOCKS]    [301,370]   7-MAY-87 12:36    1     D. KNOCKS
TT36:   [7,302]     [7,302]     7-MAY-87 10:55    0     S. ADAMS
TT37:   [GAZLE]     [303,14]    7-MAY-87 13:01    1     J. GAZLE
TT41:   [303,12]    [303,12]    7-MAY-87 12:36    0     J. ALNESE
TT42:   [304,325]   [7,325]     7-MAY-87 10:56    0     P. BEZ
TT43:   [303,23]    [303,23]    7-MAY-87 11:17    0     L. KROGAN
TT45:   [7,43]      [7,43]      7-MAY-87 14:35    0     J. MAN
VT3:    [ANGLE]     [303,5]     7-MAY-87 15:04    2     M. ANGLE
```

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's name

If Resource Acounting 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.

8.4 SHOW MEMORY

SHOW MEMORY invokes the Resource Monitoring Display (RMD), a system utility program. This program displays in a graphic manner the status of much of the system. It is, in effect, a display of the contents of the system pool (dynamic storage region).

The RMD program is useful for monitoring the general activity of the system. It is also useful viewing for a new user to see how the operating system operates. However, you should understand that the graphic display is approximate and cannot be used for critical measurement.

Format

SHOW MEMORY

The following example shows the SHOW MEMORY display for RSX–11M–PLUS. The circled numbers are keyed to the explanation.

Example

```
$ SHOW MEMORY [RET]

          ❶              ❷     ❸       ❹                ❺
 RSX-11M-PLUS V4.0 BL40 (HYSSOP)  512K   UP 000:00:49     6-JUN-87  16:25:53
 ❻TASK= PIPH1               ❼FREE= SY0:16954.  DB1:DMO
                                   DB0:8755.   DB2:DMO                PARS
❽POOL=170.:1484.:74.       ❾SECPOOL=64.:256.                         SYSPAR:D❿
      170.:1484.:74.               64.:256.:25%                       LDRPAR:D
                                                                      FCPPAR:D
   IN:      M D D     RCDMTTPSZ BPR RR RR NNF  F LNNQ B  A   P  .     SECPOL:D
   28       C I B     DORSTTOYZ YMM MM MM TTC  1 PTEM A  T   0  .     DRVPAR:D
   215K     R R O     ::::C:OSZ ERD DH DD ..S  1 0.TG P  .   0  .     GEN   :D
 ⓫OUT:      . 1 7         0 LLV V$V EA VV PNF  A  DA. 0  V   L  A ⓬
   0        . 1 F         M  01 1$3 MC 64 CSS  C  LC.    1   .  T
   OK       . M 1            G1 O$  OP    LPL  P  XP.    O   .  .
       <>!!-->    ))!)>>> >+->>->>->!!-!-->>!>->->-->!===!-+-))
   0*******32******64******96******128*****160*****192*****224*****
   E----P--D--D-----P-D----D-------------------------------------
   ---------------------------------------------------------- ⓭
   256*****288*****320*****352*****384*****416*****448*****480*****
   ->-!>->-->-><->+----+   <->-> <>---->--->                      ERRSEQ
   A I RR  A  L    .      E D   R  M   P                          1.     ⓮
   T N MM  T  0    .      D I   M  A   I
   . D DD  .  G    .      T R   D  C   P
   T F VV  V  H    E      T H   V  T   H
   3 S 71  1  0    D      3 1   5  3   1
   2 L  0  1       T      2         3
```

❶ Operating system type, version number, and base level.

❷ Six-character name that is a DECnet node name (if DEcnet is running on your system) or the system name that you selected during system generation (if DECnet is not running on your system).

❸ Size in K words of the system memory.

❹ Time elapsed in units of days, hours, and minutes since the system was last bootstrapped.

❺ Current date and time.

❻ Name of the task that is currently executing or, if none is executing, *IDLE* (Executive executing the idle loop).

❼ 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.

❽ 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.

❾ 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.

❿ 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

⓫ 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.

⓬ 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, as follows:

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.

⓭ Partition size and location. The beginning of each partition is marked with the same symbols as those listed previously plus the following additions:

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.

⓮ System error-count sequence recorded by the Error Logger (always zero if Error Logger is 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 errors in application and/or system-level software:

- Errors 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 online 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–PLUS and Micro/RSX 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. The chapter containing the main text is listed with each command. The comments are abbreviated 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 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	FORTRAN	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.

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-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual*

ANALYZE/CRASH_DUMP helps you determine the cause of system failures by analyzing and formatting a memory dump created by the Executive Crash Dump Module.

Format

ANALYZE/CRASH_DUMP[/qualifier[s]] filespec[/qualifier[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

File Qualifiers

/ACTIVE:(arg[,...])
DEVICES
TASKS
/ALL
/CACHE[:region]
/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
/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

ANALYZE/MEDIA determines if bad blocks exist on a disk volume and records their locations for use by 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
```

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 wildcard characters.
/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]] equivalencename logicalname

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=TERMINAL	!Privileged; synonym for /FINAL

ASSIGN/QUEUE *RSX-11M-PLUS and Micro/RSX System Management Guide*

ASSIGN/QUEUE establishes a path between a queue and a processor in the Queue Manager subsystem. Privileged.

Format

ASSIGN/QUEUE queuename processorname

ASSIGN/REDIRECT 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-PLUS Utilities Manual.*

BACKUP backs up and restores Files-11 volumes. It transfers a copy of files from a volume to a backup volume and retrieves the copy from the backup volume.

Format

BACKUP[/qualifier[s]] indevice:[filespec[,s]] outdevice:

Command Qualifiers

Group 1: Selective Backup and Restore

```
/CREATED:arg
          AFTER:(dd-mmm-yy hh:mm)
          BEFORE:(dd-mmm-yy hh:mm)
/EXCLUDE
/MODIFIED:arg
          AFTER:(dd-mmm-yy hh:mm)
          BEFORE:(dd-mmm-yy hh:mm)
```

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
/WINDOWS:n                  !Default is number of mapping
                            !pointers on involume.
```

Group 3: Tape 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.
/SAVE_SET:name              !Default is name of volume being
                            !backed up.
```

Group 4: Verification

```
/COMPARE
/VERIFY
```

Group 5: Display

```
/IDENTIFICATION          !Displays version of BRU
/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.
/IMAGE:arg               !For multivolume disk operations.
        SAVE
        RESTORE
/NOINITIALIZE
/LABEL:arg               !See also /LABEL:TAPE:volumelabel,
                         !Group 3.
        INPUT:volumelabel
        [OUTPUT:]volumelabel  !/LABEL:OUTPUT is default;
                         !if the only volumelabel in command line
                         !is outvolume, /LABEL:volumelabel
                         !will do.
/MOUNTED
/NEW_VERSION
/[NO]PRESERVE            !/PRESERVE is default.
/[NO]REPLACE
/SAVE_SET:name           !Default is name of volume being
                         !backed up.
```

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, only 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

Command Qualifiers

/[NO]APPEND	!/APPEND conflicts with /REPLACE.
/BLOCK_SIZE:n	!Magnetic tape block size:18 <=n <=8192; !default=512.
/[NO]FIXED_CONTROL	!/NOFIXED_CONTROL is default.
/[NO]IDENTIFICATION	!Prints RMS-11 version number; !/NOIDENTIFICATION is default; !no filespec required.
/INDEXED	!Outfile is indexed; see main text.
/KEY[:n]	!0 <=n <=9; default=1.
/[NO]LOG_FILE[:filespec]	!/NOLOG_FILE 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	!/NOTRUNCATE 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 !magnetic tape; n is octal unless !terminated with decimal point; no !effect on infile.
/[NO]CONTIGUOUS	!Specifies contiguous outfile.
/EXCLUDE:filespec	!Filespec can include wildcard characters.
/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 magnetic tape.
/REWIND	!Magnetic tape 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.

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
/OWNER_UIC:[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	!Nonoperational
/TERMINAL:ttnn:	!Privileged

DEASSIGN Chapter 5

DEASSIGN deletes logical name assignments. DEASSIGN counteracts both the ASSIGN and DEFINE commands.

Format

DEASSIGN[/qualifier[s]] logicalname

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-PLUS and Micro/RSX System Management Guide*

DEASSIGN/QUEUE counteracts ASSIGN/QUEUE. It is used to eliminate the path from a queue to a processor in the Queue Manager subsystem. Privileged.

Format

DEASSIGN/QUEUE queuename processorname

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]] logicalname equivalencename

Command Qualifiers

/FINAL	!Privileged; synonym for /TRANSLATION:FINAL
/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

/EXCLUDE:filespec	!Filespec can include wildcard characters.
/[NO]CONFIRM	!A synonym for /[NO]QUERY
/[NO]LOG	!Lists deleted files on TI:
/NOWARNINGS	!Suppresses error messages
/[NO]QUERY	!Queries before deleting;
	!/NOQUERY 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–PLUS Batch and Queue Operations Manual.*

DELETE/ENTRY deletes QMG jobs by entry number (n).

Format

DELETE/ENTRY:n[/qualifier]

Command Qualifier

/FILE_POSITION:n

DELETE/JOB Chapter 4

See also the *RSX-11M-PLUS Batch and Queue Operations Manual.*

DELETE/JOB deletes QMG jobs by queue name and job name.

Format

DELETE/JOB[/qualifier] queuename [[uic]] jobname

Command Qualifier

/FILE_POSITION:n

DELETE/processortype *RSX-11M-PLUS and Micro/RSX System Management Guide*

DELETE/processortype 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/processortype processorname

Processor Types

APPLICATIONS_PROCESSOR	
BATCH_PROCESSOR	
CARD_READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD_READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

DELETE/QUEUE *RSX-11M-PLUS and Micro/RSX System Management Guide*

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. Privileged.

Format

DELETE/QUEUE queuename/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.

Format

DIFFERENCES infile1 infile2

Command Qualifiers

/CHANGE_BAR[:n]	!n is octal ASCII code of alternate !bar. Default n=041 (!).
/IGNORE:(arg[,s])	
BLANK_LINES	
COMMENTS	!Comments begin with a semicolon.
FORM_FEEDS	
SPACING	!Group of tabs and blanks = 1 !blank.
TRAILING_BLANKS	
/LINES:n	!n lines must be same for match.
/[NO]NUMBERS	!Line numbers in output file.
/OUTPUT:filespec	!Names output file; TI: is default.
/SLP[:audittrail]	!Output is SLP indirect !command file.

DIRECTORY Chapter 4

DIRECTORY displays information on files in directories.

Format

DIRECTORY[/formatqualifier][/destinationqualifier] [otherqualifier[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 wildcard characters.
/NOWARNINGS                 !Suppresses error messages
/REWIND                     !Magnetic tape only; rewinds tape before
                            !executing command
```

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 the *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-PLUS Utilities Manual.*

EDIT/SLP invokes the Source Language Input Program (SLP), a program-maintenance editor.

Format

EDIT/SLP[/qualifier[s]] filespec

Command Qualifiers

/[NO]AUDIT[:(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 trunction of lines by audit !trail.
/[NO]TAB	!Right-justify with tabs or spaces; !/NOTAB is default.
[NO]TRUNCATE[:n]	!/TRUNCATE deletes audit trails and !trailing characters; optionally !truncates text at specified !horizontal position.

EDIT/EDI and EDIT/editor Chapter 4

EDIT/EDI invokes the Line Text Editor (EDI). EDIT/editor invokes a user-specified editor (see the following list).

Format

EDIT[/qualifier] [editinput]

Command Qualifiers

/EDI	!Line Text Editor
/KED	!Unbundled KED editor
/K52	!VT52 version of unbundled !KED editor
/MAKE	!Unsupported TECO function
/MUNG	!Unsuported 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

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,1]; g is your !group.
/LOCAL	
/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-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-PLUS Batch and Queue Operations Manual.*

HOLD/JOB holds a QMG job in its queue by queue name and job name.

Format

HOLD/JOB queuename [[uic]]jobname

INITIALIZE Chapter 5

INITIALIZE creates Files-11 disk structure on a volume. See also INITIALIZE/UPDATE. You must allocate the device and mount the volume /FOREIGN before it can be initialized. See also other forms of the INITIALIZE command following this one.

Format

INITIALIZE[/qualifier[s]] ddnn: volumelabel

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"
                                !Magnetic tape only: limits access.
                                !c can be A-Z, 0-9, space,
                                !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:;
!/NOSHOW is default.
/WINDOWS:n !Mapping pointers to file windows;
!default n=7.

INITIALIZE/processortype (Input)

RSX-11M-PLUS and Micro/RSX 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/processortype processorname[/qualifier[s]]

Processor Types

CARD_READER !INPUT is synonym.
INPUT !CARD_READER is synonym.

Command Qualifiers

/BATCH_QUEUE:queuename
/CONSOLE:ddnn:
/PRINTER_QUEUE:queuename

INITIALIZE/processortype (Output)

RSX-11M-PLUS and Micro/RSX 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/processortype 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/FORMAT Chapter 5. See also the *RSX-11M-PLUS Utilities Manual.*

INITIALIZE/FORMAT formats and verifies volumes on disk cartridges, disk packs, fixed media disks, and flexible disks associated with any RSX-11M-PLUS operating system that includes online formatting support in the Executive.

Format

INITIALIZE/FORMAT[/qualifier[s]] ddnn: volumelabel

Command Qualifiers

/BADBLOCKS
/DENSITY:arg
HIGH or DOUBLE
LOW or SINGLE
/ERROR_LIMIT:n
/MANUAL
/[NO]MESSAGE
/OVERRIDE
/[NO]VERIFY
/WRITE_LAST_TRACK:n

INITIALIZE/QUEUE *RSX-11M-PLUS and Micro/RSX System Management Guide*

INITIALIZE/QUEUE creates, names, and starts a queue in the Queue Manager subsystem.

Format

INITIALIZE/QUEUE queuename[/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 disk structure.

You must mount the volume /FOREIGN.

Format

INITIALIZE/UPDATE[/qualifier[s]] ddnn: volumelabel

Command Qualifiers

```
/ACCESSED:n
/DENSITY:arg
        LOW
        HIGH
/EXTENSION:n                !Extend full files by n blocks.
/FILE_PROTECTION:code       !Default protection for files on
                            !volume.
/LABEL:newvolumelabel       !Changes volume label.
/MAXIMUM_FILES:n            !Maximum number of files on volume.
/OWNER:[uic]
/PROFESSIONAL               !Initialize disk for Professional
                            !300 series.
/PROTECTION:code            !Protection for volume.
/[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[/qualifier[s]] [$]filespec

Command Qualifiers

```
/[NO]CHECKPOINT
/[NO]DEFER_BINDING          !/DEFER_BINDING is the default
/EXTENSION:n                !n (decimal) is additional words of
                            !address space.
/[NO]INTERPRETER            !Installing a CLI?
                            !Default is /NOINTERPRETER
/MULTIUSER_PARTITION:parname
                            !Install read-only portion.
/PARTITION:parname
/[NO]POSTMORTEM
/PRIORITY:n                 !0-250.
/[NO]READONLY_COMMON        !Install common as read-only.
                            !Default is /NOREADONLY_COMMON
/[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 information on all functions and qualifiers.

Format

LIBRARY[/operation][/qualifier[s]]
LIBRARY @filespec

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.

Format

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.

Format

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.

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.

Format

LIBRARY/REPLACE libspec 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.
        FPP                 !Floating-point processor. Default is
                            !/NOFPP.
        OTS_FAST            !Fast mapping for autoloaded
                            !memory-resident overlays
        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 command file.
/[NO]SEGREGATE                 !/NOSEGREGATE is default.
/SEQUENTIAL
/SHAREABLE[:arg]               !Multiuser.
        COMMON
        LIBRARY
        TASK                   !Default.
/SLAVE
/SLOW
/SYMBOL_TABLE[:filespec]       !Output STB file.
/[NO]SYSTEM_LIBRARY_DISPLAY
                               !Default is /NOSYS.
/[NO]TASK[:filespec]           !Names task image file;
                               !/EXECUTABLE
                               !is synonym; names 1-6 characters;
                               !/NOTASK means no task built.
/TKB                           !Default.
/TRACE
/[NO]WARNINGS                  !/NOWARNINGS 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; !/NOHOLD is default.

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-access devices); and others are limited to mounting magnetic tapes.

Format for Disks and Other Random-Access Devices

MOUNT[/qualifier[s]] ddnn: volumelabel

Format for Magnetic Tapes

MOUNT[/qualifier[s]] ddnn:[,ddnn:...] fileset-ID

Command Qualifiers for Both Disks and Tapes

```
/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
```

Command Qualifiers for Disks and Other Files-11 Devices

```
/ACCESSED:n                     !n is number of File Control
                                !Blocks.
/[NO]CACHE:([arg[,s]])          !Controls data caching; see
                                !RSX-11M-PLUS and Micro/RSX System Management Guide
           CREATE[:[region][:[mainpartition][:[size]]]]
           REGION:name
           [NO]DEFER_WRITES
           [NO]DIRECTORY[:extentsize]
           [NO]LOGICAL[:extentsize]
           [NO]OVERLAY[:extentsize]
           [NO]READ_AHEAD[:extentsize]
           [NO]VIRTUAL[:extentsize]
/EXTENSION:n                    !Extends full files by n blocks.
/OWNER:[uic]                    !Coordinates with file and volume
                                !protection.
/UNLOCK                         !Main use is with VFY.
```

Command Qualifiers

```
/WINDOW:arg
        n                       !Default for n is value specified when
        USER:n,INDEX:n          !volume was initialized;
                                !n can be 1 to 127
        FULL
```

Command Qualifiers for ANSI and Unlabeled Tapes

```
/BLOCK_SIZE:n
/CARRIAGE_CONTROL:arg
                        FORTRAN
                        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-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

```
/[NO]ADJACENT
/AFTER:(dd-mmm-yy hh:mm)
/AFTER:TOMORROW[: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           !Flag page on job; default is
                        !/JOB_PAGE.
/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:queuename
/[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 wildcard characters.
/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-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

RELEASE/JOB Chapter 4

See also the *RSX-11M-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 queuename [[uic]]jobname

REMOVE Chapter 7

REMOVE counteracts INSTALL. REMOVE takes a task name out of the System Task Directory. Privileged.

Format

REMOVE[/qualifier] taskname

Command Qualifiers

/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 wildcard characters in any of the file specification fields, but please exercise care in using them.

Format

RENAME[/qualifier[s]] oldfilespec newfilespec

Command Qualifiers

/EXCLUDE:filespec	!Filespec can include wildcard characters.
/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

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	
/COMMAND:"taskcommand"	!See main text.
/EXTENSION:n	!n (decimal) additional words of !address space.
/[NO]IO_PAGE	

/PARTITION:parname	!Privileged
/[NO]POSTMORTEM	
/PRIORITY:n	!Privileged
/READ_PARTITION:pname	
/[NO]SLAVE	
/STATUS:arg	
TASK	!Return status from task being run; !the default
COMMAND	!Return status from RUN command; !see main text on both these !arguments.
/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

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

```
/[NO]CACHE:(arg[,s])              !Modifies data caching; see
                                  !RSX-11M-PLUS and Micro/RSX System Management Guide.
        CREATE[:[region][:[mainpartition][:[size]]]]
        REGION:name
        [NO]DEFER_WRITES
        [NO]DIRECTORY[:extentsize]
        [NO]LOGICAL[:extentsize]
        [NO]OVERLAY[:extentsize]
        [NO]READ_AHEAD[:extentsize]
        [NO]VIRTUAL[:extentsize]
/[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)
                                  !See main text
/ENTER:synonym_filespec           !See main text
/NOWARNINGS                       !Suppresses error messages
/REMOVE                           !See main text
/REWIND                           !Magnetic tape only; rewinds tape before
                                  !beginning operation
/TRUNCATE                         !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

Command Qualifiers

/DTE

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 to 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 to 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)

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 wildcard characters.

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-11M-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-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.

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

/RELEASE !Same as RELEASE/QUEUE
/[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[:ddnn:/REGISTER=nnnnnn] !Loads a crash driver.
/DIRECTORY:[directory] !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
!(EXTK$) directive.
/[NO]LOGINS
/NETWORK_UIC
/PACKETS:n !n = 0 through 15.
/POOL:top:max:total !Increases size of pool; see main
!text.
/POOL/LIMITS:arg
HIGH=n
LOW=n
MINIMUM_SIZE:n
TASK_PRIORITY:n

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]

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	
/LA12	
/LA30P	
/LA30S	
/LA34	
/LA36	
/LA38	
/LA50	
/LA75	
/LA100	
/LA120	
/LA180S	
/LA200_SERIES	
/LA210	
/LFFILL	
/LN03	
/LQP01	
/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

/[NO]TTSYNC
/VT05B
/VT50
/VT52
/VT55
/VT61
/VT100
/VT101
/VT102
/VT105
/VT125
/VT131
/VT132
/VT200_SERIES
/WIDTH:n

Group 3: Task Setup

/CHARACTER_LENGTH:n
/[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]TYPE_AHEAD[: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

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-PLUS and Micro/RSX 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.

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

Command Qualifier

/VERSION

SHOW LIBRARY Chapter 8

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]

Command Qualifier

/DIRECTORY	!Nonoperational, !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 LOGICALS.

Format

SHOW LOGICALS[/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 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-PLUS Batch and Queue Operations Manual.*

SHOW PROCESSOR displays information about the processors, batch processors, printers, card readers, and other devices controlled by the Queue Manager.

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 Program (ACNT) to enter a protection code for your account.

Format

SHOW PROTECTION

SHOW QUEUE Chapter 4

See also the *RSX-11M-PLUS Batch and Queue Operations Manual.*

SHOW QUEUE displays information about batch and print jobs in queues.

Format

SHOW QUEUE[/qualifier] [queuename]

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.

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 where 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/ACTIVE/DYNAMIC Chapter 7

SHOW TASKS/DYNAMIC invokes RMD to display on a video terminal continuously updated information on the activity of tasks on the Active Task List. On a hardcopy terminal, SHOW TASKS/ACTIVE/DYNAMIC provides a static 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.

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 !change; 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
/CHARACTER_LENGTH:n
/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]LA12
/[NO]LA30P
/[NO]LA30S
/[NO]LA34
/[NO]LA36
/[NO]LA38
/[NO]LA50
/[NO]LA75
/[NO]LA100
/[NO]LA120
/[NO]LA180S
/[NO]LA200_SERIES
/[NO]LA210

```
/[NO]LFFILL
/[NO]LN03
/[NO]LOCAL
/LOGGED_ON
/[NO]LQP01
/[NO]LQP02
/[NO]LQP03
/[NO]LOWERCASE          !/NOLOWERCASE is same as /UPPERCASE
                        !and is default.

/MCR
/MODEL
/PAGE_LENGTH
/[NO]PARITY
/[NO]PASSALL
/[NO]PASTHRU
/[NO]PRIVILEGE
/PRINTER_PORT
/[NO]PRO_SERIES
/[NO]REGIS
/[NO]REMOTE
/RT                     !DECnet host terminal
/[NO]SCOPE
/[NO]SERIAL
/[NO]SLAVE
/[NO]SOFT_CHARACTERS
/SPEED
/[NO]TAB
/TI
/TT                     !All real terminals
/[NO]TTSYNC
/[NO]TYPEAHEAD
/[NO]UPPERCASE          !/NOUPPERCASE is same as
                        !/LOWERCASE.
/VT                     !Virtual terminal
/[NO]VT05B
/[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

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 (STOP$S) directive. The task name defaults to TTnn:.

Format

START[/qualifier] [taskname]

Command Qualifier

/TERMINAL:ttnn:	!Privileged

START/processortype *RSX-11M-PLUS and Micro/RSX System Management Guide*

This form of the START command starts a processor, batch processor, output processor, or card-reader processor. Privileged.

Format

START/processortype processorname[/qualifier[s]]

Processor Types

APPLICATIONS_PROCESSOR	
BATCH_PROCESSOR	
CARD_READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD_READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

Command Qualifiers

/FORMS:n	!Overrides value set on !initialization.
/CONTINUE	!Default
/RESTART	
/NEXT	
/TOP_OF_FILE	
/BACKSPACE:n	
/FORWARDSPACE:n	
/PAGE:n	
/ALIGN	

START/QUEUE *RSX-11M-PLUS and Micro/RSX System Management Guide*

START/QUEUE starts a queue. Privileged.

Format

START/QUEUE queuename

START/QUEUE/MANAGER *RSX-11M-PLUS and Micro/RSX 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

STOP/processortype *RSX–11M–PLUS and Micro/RSX 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/processortype processorname[/qualifier[s]]

Processor Types

APPLICATIONS_PROCESSOR	
BATCH_PROCESSOR	
CARD_READER	!INPUT is synonym.
DEVICE	!PRINTER is synonym.
INPUT	!CARD_READER is synonym.
PRINTER	!DEVICE is synonym.
PROCESSOR	

Command Qualifiers

/ABORT	!See main text.
/FILE_END	
/JOB_END	
/PAUSE	

STOP/QUEUE *RSX–11M–PLUS and Micro/RSX System Management Guide*

STOP/QUEUE stops queues. Privileged.

Format

STOP/QUEUE queuename

STOP/QUEUE/MANAGER *RSX-11M-PLUS and Micro/RSX 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-PLUS Batch and Queue Operations Manual*

SUBMIT queues batch jobs for processing by a batch processor.

Format

SUBMIT[/qualifier[s]] filespec[,s]

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 !file name.
/[NO]PRINT[:queuename]	!Optionally names 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 wildcard characters.
/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	

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 wildcard characters.
/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 program (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 letters A to Z, the numerals 0 to 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.

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 the 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 the 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 prepare 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.

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: Reenter the command after checking proper syntax.

yyy—Command line incomplete

Explanation: The command as typed is not a complete command.

User Action: Reenter the command after checking proper syntax.

yyy—Command syntax error

Explanation: The command did not conform to the syntax rules.

User Action: Check the 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: Reenter the command after checking proper syntax.

yyy—Contradictory qualifier

Explanation: The command as typed included contradictory qualifiers, such as /DELETE and /NODELETE.

User Action: Check for proper syntax and reenter the command.

yyy—Contradictory qualifier in key specification

Explanation: The command included a contadictory qualifier in the key definition argument to the /KEY qualifier.

User Action: Check for proper syntax and reenter the command.

yyy—Decimal number expected

Explanation: The command included a number not in proper format.

User Action: Check the command for proper syntax and reenter with a decimal point (.) terminating the number.

yyy—Destination expected

Explanation: The command line did not include the destination parameter.

User Action: Reenter the command line after checking for proper syntax.

yyy—Device invalid or not specified

Explanation: The command specified an invalid device or no device at all when a device name is required.

User Action: Check the devices on the system with SHOW DEVICES. Reenter the 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 the command after checking for proper syntax.

yyy—Device not mounted/allocated

Explanation: The command specified a device that is not properly mounted or allocated for the command to execute.

User Action: Check the status of the device with SHOW DEVICES. Find the cause of the error and correct it if you can. It is likely that you will need to mount the volume on the device. Nonprivileged users need the volume label to mount the volume.

yyy—Device not terminal

Explanation: The command specified a device other than a terminal where a terminal device name is required. TI: is not an acceptable terminal device name in all contexts.

User Action: Reenter the command after checking for proper syntax.

yyy—Directory write protected

Explanation: The command attempted to remove an entry from a directory that is privileged or from a directory on a device that is write-protected.

User Action: Determine the cause of the error and correct it if you can. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

yyy—Error dispatching commands. DSW = "n"

Explanation: An error occurred that was not explicitly handled by DCL or some invoked task.

User Action: Determine the meaning of the DSW error code in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual.* Determine the cause of the error and correct it if you can. If you cannot, see your system manager or operator.

yyy—Explicit output file name required

Explanation: The command requires an explicit output file name.

User Action: Reenter the command line in proper syntax and without wildcards.

yyy—Extraneous input—Check for proper command structure

Explanation: The command as typed included extraneous input. The circumflex (^) points to the error or just past the last successfully parsed command element.

User Action: Reenter the command line in proper syntax.

yyy—Failed to attach output device

or

yyy—Failed to detach output device

Explanation: An attempt to attach or detach a record-oriented output device, such as a terminal or line printer, failed. This error usually means the device is off line or nonresident.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

yyy—Failed to create output UFD

Explanation: The command failed to create an entry in a directory because the device was write-protected or because of a privilege violation.

User Action: Determine the cause of the error and correct it if you can. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

yyy—Failed to delete file

or

yyy—Failed to mark file for delete

Explanation: The command attempted to delete a protected file.

User Action: Check for the proper file specification and default 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 want to affect. You may need to check the validity of the volume with the VFY utility.

yyy—Failed to remove directory entry

Explanation: The command attempted to remove an entry from a directory that was either protected against access or on a write-protected device.

User Action: Determine the cause of the error and correct it if you can. You may need to enable write access through the device hardware or change the protection for the directory. If you cannot correct the error, see your system manager or operator.

yyy—Failed to truncate file

Explanation: The command specified a volume that is corrupted or is protected against access.

User Action: You may be able to correct the error by making your defaults the same as the device and directory of the file you want 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 want 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: Reenter the command after checking for proper syntax.

yyy—File not locked

Explanation: An UNLOCK command specified a file that is not locked.

User Action: Check for the proper file specification and reenter the command.

yyy—File specification either invalid or not specified

Explanation: The system could not read a file specification included in the command. This error often results from a typing mistake or typing the command in the wrong format.

User Action: Check for proper syntax and reenter the command.

yyy—File specification list not available for RMS-11

Explanation: A command to an RMS-11 utility included more than one input file specification.

User Action: Check for proper syntax and reenter the command.

yyy—File version number not specified

Explanation: The command requires a file version number to be specified.

User Action: Reenter the command after checking for 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 invalid indirect command file.

User Action: Check for a 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 the 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 invalid format or contained some other syntax error.

User Action: Check the command for proper syntax and reenter. Device names are two alphabetic characters followed by an octal number and a colon.

yyy—Illegal filespec

Explanation: The command required a file specification that was not present.

User Action: Check the 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 the 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 the 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 the 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 the 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 the command for proper syntax and reenter. See Chapter 4 for more information on protection codes.

yyy—Illegal qualifier value

Explanation: The command as typed included an improper argument to a qualifier.

User Action: Check the 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 the 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 invalid format.

User Action: Check the 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 the 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 SHOW ASSIGNMENTS), correct the definition, and reenter the command line.

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 the 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 the 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 the command for proper syntax and reenter.

yyy—Invalid device

Explanation: The SET DEFAULT command was incorrectly issued without parameters.

User Action: Reenter the command with proper syntax.

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: Reenter the command after checking proper syntax.

yyy—Invalid file specification qualifier

Explanation: The command included a qualifier to a file specification that was not valid. In DCL, a qualifier is preceded by a slash (/).

User Action: Check for proper syntax and retype the command.

yyy—Invalid terminal specified

Explanation: A command directed to a specific terminal named the terminal in an improper format.

User Action: Check the name of the terminal using SHOW USERS and reenter the command with the proper format.

yyy—Invalid time or date

Explanation: The command specified a clock or calendar field, or both, incorrectly.

User Action: Check for proper syntax and retype the command.

yyy—Invalid UIC specified

Explanation: The command specified a 0, a number including an 8 or 9, or a number greater than 377_8 for the group or member number in a UIC or directory.

User Action: Reenter the command after checking for proper syntax.

yyy—I/O error on input file

or

yyy—I/O error on output file

Explanation: One of the following conditions exists:

- The device is not on line.
- The device is not mounted.
- The hardware has failed.
- The output volume is full.
- The input file is corrupted.

User Action: Determine the cause of the error and correct it if you can. If you cannot, see your operator or system manager.

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: Reenter the command after you have checked for proper syntax.

yyy—More command parameters than permitted

Explanation: The command as typed included too many parameters.

User Action: Check the 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 Program (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.

yyy—Numeral expected

or

yyy—Numeral required

Explanation: The command included nonnumeric characters in a position in which numerals are required or expected.

User Action: Check the 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 the 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 the 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.

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 FORTRAN 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 the 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 the 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 letters A to Z, the numerals 0 to 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: Reenter the command after checking for proper syntax.

yyy—Repeated key specification

Explanation: The command as typed included the same key specification more than once in the key definition argument to the /KEY qualifier.

User Action: 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 the 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 the command after checking the syntax to see if it is correct.

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: Reenter the command after checking for proper syntax.

yyy—Required value not specified for position size or number

Explanation: The command failed to include a required value in the key definition argument to the /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 the 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 invalid element, as identified by the quoted part of the message.

User Action: Check the 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 database. 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 command, 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 TASKS commands or with SHOW MEMORY.

yyy—Sorry, task not installed

Explanation: The command attempted to invoke a task that was not installed.

User Action: Try running the task with a command in the RUN $ form. Determine the cause of the error and correct it if you can. If you cannot, see your system manager or operator.

yyy—Syntax error

Explanation: The command included some error in typing or specification, such as a letter where a number should appear.

User Action: Check for proper syntax and reenter the 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: The command included a wildcard (* or %) in a context where it is not permitted.

User Action: Check for proper syntax and reenter the command.

10.2 Task Termination Messages

The following messages are issued by the Task Termination Notification program (TKTN). These messages appear as part of the notification when a task is aborted. Often, a list of register contents also is displayed. Understanding these messages requires an understanding of how the PDP-11 processor 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 the command for the 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.

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-protection 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 invalid instruction, and no SST routine was specified to process the trap.

SST abort. Bad stack

Explanation: An SST could not be effected because the SST parameters could not be pushed onto the task's stack.

Task exit with outstanding IO

Explanation: The task exited with one or more outstanding I/O requests. Tasks should terminate all I/O operations before exiting. The system does, however, clean up all outstanding I/O.

T bit trap or BPT instruction

Explanation: The task either set the T-bit in the Processor Status Word or executed a BPT instruction, and no SST routine was specified to process the trap.

Trap execution

Explanation: The task executed a Trap instruction, and no SST routine was specified to process the trap.

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	
COBOL	COB	

Table A–1 (Cont.): Functions Initiated by DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
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 assignment 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	
EDIT	EDT, EDI, KED, SLP	
FORTRAN	FOR, F4P, F77	
FIX	FIX	
HELP	HELlo	As part of HELlo, HELP can run on unlogged-in terminals.

Table A–1 (Cont.): Functions Initiated by DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
HOLD/ENTRY	QMG	
HOLD/JOB	QMG	
INITIALIZE	INItvolume	Create volume
INITIALIZE/FORMAT	FMT	Format disk
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	
RENAME	PIP	
REQUEST	BROadcast	
RUN	RUN	
SET [DAY]TIME	TIMe	
SET DEFAULT	SET /UIC	SET for UFD ASN for SY:
SET DEVICE	DEVice	

Table A–1 (Cont.): Functions Initiated by DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
SET FILE	PIP	
SET GROUPFLAGS	FLAg	
SET HOST	SET HOST	
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	
SHOW COMMON	CBD	Common Block Directory
SHOW [DAY]TIME	TIMe	
SHOW DEFAULT	SET /UIC	SET for UFD
	ASN	ASN for SY:
SHOW DEVICES	DEV	
	SET /PUB	/PUBLIC qualifier
	SET /BUF	/WIDTH qualifier

Table A–1 (Cont.): Functions Initiated by DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
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	
SHOW TERMINAL	DEV	
SHOW TERMINAL/attribute	SET /attribute	
SHOW UIC	SET/UIC	
SHOW USERS	DEV	
SORT	SRT	SORT–11
START	UNS	
START/PROCESSOR	QMG	
START/QUEUE	QMG	
START/QUEUE/MANAGER	QMG	

Table A–1 (Cont.): Functions Initiated by DCL Commands

DCL Command	MCR Command, Task, or Utility	Comments
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

A

ABORT command, 7-17 to 7-20, 9-2
Account File Maintenance Program
  See ACNT
ACNT, 3-9 to 3-11, 4-1
ACP, 5-4, 5-49
ALLOCATE command, 5-10, 5-11, 5-41 to 5-43, 9-2
ANALYZE command
  ANALYZE/CRASH_DUMP, 9-2 to 9-3
  ANALYZE/MEDIA, 5-11, 5-18 to 5-23, 9-3
Ancillary Control Processor
  See ACP
ANSI file name, 1-15
APPEND command, 4-63 to 4-64, 9-4
ASSIGN command, 5-15, 5-24 to 5-30, 9-4 to 9-5
  ASSIGN/QUEUE, 9-5
  ASSIGN/REDIRECT, 5-31 to 5-33, 9-5
  ASSIGN/TASK, 7-42 to 7-43, 9-5

B

BACK SPACE key, 3-13
Backup and Restore Utility
  See BACKUP command
BACKUP command, 5-91 to 5-130, 9-5 to 9-7
  backing up
    Files–11 volume, 5-92
    foreign volume, 5-92
  /INITIALIZE qualifier, 5-92
  /MOUNTED qualifier, 5-92
  restoring
    Files–11 volume, 5-92
    foreign volume, 5-92
Bad block, 5-11, 5-18 to 5-23, 5-68
Batch job
  stopping, 4-92
Batch processing, 1-23
Bell, 3-22
BROADCAST command, 3-32 to 3-35, 9-7
BRU
  See BACKUP command

C

CANCEL command, 7-23, 9-7
CAPS LOCK key, 3-14
CLI, 1-1
Clock queue, 7-13 to 7-16, 7-23, 9-41
  deleting entry, 7-23
  displaying, 7-72
Command
  abbreviating, 1-12, 1-24
  MCR, 1-12
Command file, 6-5 to 6-6
Command line, 1-8
  comments, 1-20
  continuation, 1-20
  length, 3-12
Command line interpreter
  See CLI
Common
  displaying, 7-73 to 7-74
  status flags, 7-73 to 7-74
CONTINUE command, 7-21 to 7-22, 9-8
Control key
  See CTRL key
CONVERT command, 4-53 to 4-58, 9-8
COPY command, 4-45 to 4-49, 9-9
COPY key, 3-15
CREATE command, 4-14 to 4-15, 9-9
  CREATE/DIRECTORY, 4-16 to 4-18, 9-10

CTRL key, 3-18
CTRL/C, 3-19, 7-18 to 7-20
CTRL/G, 3-22
CTRL/I, 3-22
CTRL/K, 3-22
CTRL/L, 3-21
CTRL/M, 3-22
CTRL/O, 3-20
CTRL/Q, 3-19
CTRL/R, 3-21
CTRL/S, 3-19
CTRL/U, 3-20
CTRL/X, 3-21
CTRL/Z, 3-20

D

SET command
SET [DAY]TIME, 9-32
SHOW command
SHOW [DAY]TIME, 9-41
Date, 1-13
displaying, 8-4
setting, 8-2
DCL, 1-1, 9-1
DEALLOCATE command, 5-44 to 5-45, 9-10
DEASSIGN command, 5-34 to 5-36, 9-10
DEASSIGN/QUEUE, 9-11
DEBUG command, 7-44 to 7-45, 9-11
DECnet, 1-15
DEFINE command, 5-14, 5-37 to 5-40, 9-11
DELETE command, 4-38 to 4-41, 9-12
DELETE/DIRECTORY, 4-19, 9-12
DELETE/ENTRY, 4-83 to 4-84, 9-12
DELETE/JOB, 4-83 to 4-84, 9-13
DELETE/PROCESSOR, 9-13
DELETE/QUEUE, 9-13
DELETE key, 1-22, 3-13
Device, 5-1
See also Pseudo device, Logical device
allocated, 5-10
displaying attributes, 5-138
industrial, 5-5
laboratory, 5-5
logical, 5-6, 5-36
peripheral, 5-1 to 5-10
private, 5-41 to 5-45
public, 5-13
setting attributes, 5-134
unowned, 5-10
DIFFERENCES command, 4-65 to 4-70, 9-14
DIGITAL Command Language
See DCL
DIGITAL Standard Editor
See EDT editor
Directory, 4-1, 4-3
creating, 4-16
default, 4-4, 4-96, 4-100
deleting, 4-19
library, 7-28 to 7-31, 8-10
displaying, 8-11
setting, 8-10
maintaining, 4-29
named, 4-96
numbered, 4-96
protection, 4-12 to 4-13, 4-17, 4-107 to 4-109
default, 4-110 to 4-112
system, 7-28 to 7-31, 8-10
DIRECTORY command, 4-29 to 4-37, 9-14
Disk
blank, 5-11 to 5-12
new, 5-11 to 5-12
scratch, 5-11 to 5-12
DISMOUNT command, 5-12, 5-13, 5-63 to 5-66, 9-15

E

EDI editor, 4-26
EDIT command, 4-20 to 4-28
EDIT/EDI, 4-26 to 4-28, 9-16
EDIT/EDT, 4-20 to 4-23
EDIT/SLP, 4-24 to 4-25, 9-16
/RECOVER qualifier, 4-21
EDT editor, 4-20
ENTER key, 3-13
Equal sign (=), 1-8
Error messages
DCL, 10-1 to 10-16
ESCAPE key, 3-13
Event flag
displaying, 7-77
setting, 7-75 to 7-76

F

File
appending, 4-63
attributes
setting, 4-101
comparing, 4-65
copying, 4-45
creating, 4-14, 4-20

File (cont'd.)
deleting, 4-38
editing, 4-20
recovering, 4-21
library, 6-4
location, 4-1
object, 6-3, 6-8
organization, 4-1
overlay, 6-5
owner, 4-1
printing, 4-73
protection, 9-35
purging, 4-42
renaming, 4-50
source, 6-2
editing, 4-24
task image, 6-3, 6-8
typing, 4-60
unlocking, 4-71
version number, 4-3
default, 4-4
File name, 4-3
File protection, 2-7, 4-9 to 4-12
See also Protection
default, 4-113
setting, 4-107 to 4-109, 4-110 to 4-112
Files–11, 5-4
volume
initializing, 5-67
File specification, 4-2 to 4-4
defaults, 4-4 to 4-6
format, 4-2
File type, 4-3
CMD, 6-6
default, 4-4, 4-5 to 4-6
OBJ, 6-6
ODL, 6-5
FIX command, 7-39 to 7-40, 9-17
Flag page, 4-75
/FOREIGN qualifier
MOUNT command, 5-10, 5-11, 5-13, 5-48
Form feed, 3-21

H

HELP command, 1-10 to 1-12, 3-23 to 3-26, 9-17
Help file, 3-26 to 3-31
writing, 3-26
HOLD command
HOLD/ENTRY, 4-93, 9-17
HOLD/JOB, 4-93, 9-18
HOLD SCREEN key, 3-15

I

Indirect command file, 1-23
BACKUP command, 5-124
INITIALIZE command, 5-4, 5-11, 5-13, 5-67 to 5-79, 9-18
INITIALIZE/FORMAT, 5-80 to 5-86, 9-20
INITIALIZE/PROCESSOR
input, 9-19
output, 9-19
INITIALIZE/QUEUE, 9-20
INITIALIZE/UPDATE, 5-87 to 5-90, 9-20
/INQUIRE qualifier
SET TERMINAL command, 3-41
INSTALL command, 7-28 to 7-36, 9-21

K

Keyboard, 3-12 to 3-17
keys table, 3-15 to 3-17
Keypad, 3-17 to 3-18
alternate, 3-17 to 3-18
editing
VT100-series terminal, 3-17
VT200-series terminal, 3-17

L

Language
programming, 6-1
source, 6-1
Library, 6-32 to 6-34
macro, 6-33
object module, 6-33
universal, 6-33
LIBRARY command, 6-32 to 6-56, 9-22
LIBRARY/COMPRESS, 6-35 to 6-36, 9-22
LIBRARY/CREATE, 6-37 to 6-39, 9-22
LIBRARY/DELETE, 6-40, 9-23
LIBRARY/EXTRACT, 6-41 to 6-42, 9-23
LIBRARY/INSERT, 6-43 to 6-44, 9-23
LIBRARY/LIST, 6-45 to 6-48, 9-23
LIBRARY/REMOVE, 6-49, 9-24
LIBRARY/REPLACE, 6-50 to 6-51, 9-24
Library directory, 7-28 to 7-31
displaying, 8-11
setting, 8-10
Line continuation, 1-20
Line terminator, 3-13
Line Text Editor
See EDI editor

LINK command, 6-5, 6-7 to 6-31, 9-24 to 9-26
  option, 6-26 to 6-28
Logical device, 5-6, 5-9 to 5-10, 5-24 to 5-30, 5-34 to 5-36
  assigning, 5-9, 5-24 to 5-30
  deleting, 5-34 to 5-36
  displaying, 5-130 to 5-133
Logical device name, 5-9 to 5-10
Logical name, 5-6
  assigning, 5-24 to 5-30
  defining, 5-37 to 5-40
  deleting, 5-34 to 5-36
  described, 5-14 to 5-17
  displaying, 5-15, 5-130 to 5-133
  format, 5-15
  translation, 5-16
Logical name table, 5-14
Logical unit number
  See LUN
LOGIN command, 3-4 to 3-7, 9-26
  password, 3-4
  UIC, 3-4
LOGOUT command, 3-8 to 3-9, 5-14, 7-20, 9-26
LUN, 5-6, 5-8 to 5-9, 7-42 to 7-43, 7-68 to 7-69

M

Macro library, 6-33
Magnetic tape, 5-13
  blank, 5-13
  mounting, 5-58 to 5-62
  new, 5-13
Magtape
  See Magnetic tape
Master File Directory
  See MFD
MCR, 1-1 to 1-2
MCR command, 9-26
MFD, 4-1, 4-2, 5-11
Monitor Console Routine
  See MCR
MOUNT command, 5-4, 5-12, 5-13, 5-46 to 5-62, 9-26 to 9-28
  qualifiers
    /FOREIGN, 5-10, 5-11, 5-13, 5-48
    /[NO]SHAREABLE, 5-13, 5-50
    /NOLABEL, 5-55

N

Network, 1-15
/[NO]SERIAL qualifier
  SET TERMINAL command, 3-48
/[NO]SHAREABLE qualifier
  MOUNT command, 5-50
Node, 1-15
/NOLABEL qualifier
  MOUNT command, 5-55
NO SCROLL key, 3-15

O

Object file, 6-4
Object module library, 6-33

P

Parameter
  displaying system, 8-8 to 8-9
  setting system, 8-5 to 8-7
  syntax, 1-14
Partition
  system memory, 7-46 to 7-52
Password, 3-4
  changing, 3-10
Pool
  displaying, 8-14 to 8-17
PRINT command, 4-73 to 4-82, 9-28
Printer
  stopping, 4-92
Print job
  stopping, 4-92
Priority
  task, 7-70 to 7-71
Program
  compiling, 6-1
  developing, 6-1 to 6-31
Prompt, 1-8
Protection, 4-9 to 4-13, 4-107 to 4-113, 9-35
  directory, 4-12 to 4-13, 4-17
  file, 4-9 to 4-12, 5-98, 9-35
    default, 4-113
    setting, 4-107 to 4-109, 4-110 to 4-112
  volume, 4-13, 5-69
Pseudo device, 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

Pseudo device (cont'd.)
SY:, 5-7
TI:, 3-1, 5-7, 5-8
VT:, 5-7, 5-8
WK:, 5-7, 5-8
PURGE command, 4-42 to 4-44, 9-29

Q

QMG, 1-23, 4-83 to 4-94
entry number, 4-83
Qualifier, 1-9
abbreviating, 1-12
date-related, 4-8
Queue Manager
See QMG
Quotation mark ("), 1-14

R

Record
moving, 4-53
/RECOVER qualifier
EDIT command, 4-21
RELEASE command
RELEASE/ENTRY, 4-94, 9-30
RELEASE/JOB, 4-94, 9-30
REMOVE command, 7-37 to 7-38, 9-30
RENAME command, 4-50 to 4-52, 9-30
REPEAT key, 3-14
REQUEST command, 3-36 to 3-37, 9-31
Resource Accounting, 3-8, 3-11, 9-40
RETURN key, 3-12 to 3-13, 3-22
RSX–11M–PLUS and VMS, 1-24
RUN command, 7-6 to 7-16, 9-31 to 9-32
installed task, 7-12 to 7-15
uninstalled task, 7-8 to 7-12

S

Scratch disk
formatting, 5-11
SET and SHOW commands, 1-3
SET command, 2-1
SET [DAY]TIME, 2-2, 8-2, 9-32
SET DEBUG, 1-20, 9-32
SET DEFAULT, 2-6, 4-96 to 4-99, 9-32
SET DEVICE, 5-134 to 5-137, 9-33
SET FILE, 4-101 to 4-103, 9-33
SET GROUPFLAGS, 7-75 to 7-76, 9-34
SET HOST, 1-17, 9-34
SET LIBRARY/DIRECTORY, 8-10, 9-34
SET PARTITION, 7-46 to 7-49, 9-34
SET PASSWORD, 3-10 to 3-11, 9-35
SET PRIORITY, 7-70 to 7-71, 9-35
SET PROTECTION, 4-107 to 4-109, 9-35
SET PROTECTION/DEFAULT, 4-110 to 4-112, 9-35
SET QUEUE, 4-89 to 4-91
SET QUEUE/ENTRY, 9-36
SET QUEUE/JOB, 9-36
SET SYSTEM, 8-5 to 8-7, 9-37
SET TERMINAL, 9-37 to 9-39
SET UIC, 4-104 to 4-105, 9-39
SET TERMINAL command, 3-19, 3-38 to 3-50
qualifiers
/INQUIRE, 3-41
/[NO]CONTROL, 3-40
/[NO]SERIAL, 3-48
/SHAREABLE qualifier
MOUNT command, 5-13
SHIFT key, 3-14
SHOW command, 2-1
SHOW [DAY]TIME, 2-2, 8-4, 9-41
SHOW ACCOUNTING, 9-40
SHOW ASSIGNMENTS, 2-5, 5-15, 5-130 to 5-133, 9-40
SHOW CACHE, 5-142, 9-40
SHOW CLOCK_QUEUE, 7-72, 9-40
SHOW COMMON, 7-73 to 7-74, 9-41
SHOW DEFAULT, 4-100, 9-41
SHOW DEVICES, 5-5, 5-138 to 5-141, 9-41
SHOW GROUPFLAGS, 7-77
SHOW HOST, 1-19, 9-42
SHOW LIBRARY, 8-11, 9-42
SHOW LOGICALS, 2-5, 5-15, 5-130 to 5-133, 9-42
SHOW MEMORY, 2-2, 8-14 to 8-17, 9-43
SHOW PARTITIONS, 7-50 to 7-52, 9-43
SHOW PROCESSOR, 2-5, 4-88, 9-43
SHOW PROTECTION, 4-113, 9-43
SHOW QUEUE, 2-4, 4-85 to 4-87, 9-44
SHOW SYSTEM, 8-8 to 8-9, 9-44
SHOW TASKS, 7-53 to 7-69, 9-45
SHOW TASKS/ACTIVE, 7-54 to 7-59
SHOW TASKS/INSTALLED, 7-60 to 7-61
SHOW TASKS/INSTALLED/DEVICE, 7-62
SHOW TASKS/LOGICAL_UNITS, 7-68 to 7-69
SHOW TERMINAL, 3-51 to 3-59, 9-46 to 9-47
display, 2-3

SHOW command (cont'd.)
SHOW UIC, 4-106, 9-48
SHOW USERS, 8-12 to 8-13, 9-48
SHOW TASKS command
SHOW TASKS/ACTIVE/DYNAMIC, 7-63 to 7-65, 9-45
SHOW TASKS:taskname/DYNAMIC, 7-66 to 7-67, 9-45
SLP, 4-24
Source Language Input Program
See SLP
START command, 7-26, 9-48
START/PROCESSOR, 9-48, 9-49
START/QUEUE, 9-49
START/QUEUE/MANAGER, 9-49
START/UNBLOCK, 7-27, 9-49
Status flags
common, 7-73 to 7-74
task, 7-56 to 7-58
STD, 7-1 to 7-5
STOP command
STOP/ABORT, 4-92, 9-49
STOP/BLOCK, 7-24 to 7-25, 9-50
STOP/PROCESSOR, 9-50
STOP/QUEUE, 9-50
STOP/QUEUE/MANAGER, 9-51
SUBMIT command, 9-51
Symbol
TKB, 6-7
System
displaying parameters, 8-8 to 8-9
setting parameters, 8-5 to 8-7
System directory, 7-28 to 7-31, 8-10
System failure, 9-2 to 9-3
System status
displaying, 8-14 to 8-17
System Task Directory
See STD

T

Tab, 3-22
vertical, 3-22
Task, 7-1
aborting, 7-17 to 7-20
active, 7-1 to 7-2
displaying, 7-53 to 7-59
dynamic display, 7-63 to 7-67
blocking, 7-24 to 7-25
continuing, 7-21 to 7-22
dormant, 7-1
executing, 7-6 to 7-16
Task (cont'd.)
fixing, 7-39 to 7-40
installed, 7-1 to 7-2, 7-3, 7-12 to 7-16
displaying, 7-60 to 7-61
installing, 7-28 to 7-36
priority
setting, 7-70 to 7-71
removing, 7-37 to 7-38
starting, 7-26
unblocking, 7-27
Task Builder
See TKB
Task image file, 7-8 to 7-12
Task name, 7-2 to 7-5
Task Termination Notification program
See TKTN
Terminal, 3-1 to 3-11
attached, 3-2
characteristics
displaying, 3-51
setting, 3-38, 9-37
showing, 9-46
slaved, 3-2
VT100-series, 3-17
special keys, 3-18
VT200-series, 3-17
special keys, 3-18
VT52, 3-17
special keys, 3-18
Time
displaying, 8-4
setting, 8-2
TKB, 6-1, 6-6 to 6-7
input file, 6-3 to 6-6
symbol, 6-7
TKTN
messages, 10-17 to 10-18
TYPE command, 4-60 to 4-62, 9-51 to 9-52

U

UFD
See Directory
UIC, 4-1, 4-104 to 4-106
Underscore character (_), 1-14
UNFIX command, 7-41, 9-52
UNLOCK command, 4-71 to 4-72, 9-52
User File Directory (UFD)
See Directory

User Identification Code
  See UIC
Users
  displaying list of, 8-12 to 8-13

V

Version number
  file, 4-3
    default, 4-4
VMS DCL, 1-24
Volume, 4-1, 5-1 to 5-10
  backing up, 5-91
  Files–11, 5-4
    initializing, 5-67
  label, 1-15
  mounted, 5-10
  mounting, 5-46 to 5-62
  private, 5-10, 5-41 to 5-45, 5-50
  public, 5-10, 5-50
  restoring, 5-91
  shareable, 5-10, 5-50
VT100-series terminal, 3-17
  special keys, 3-18
VT200-series terminal, 3-1, 3-17
  special keys, 3-18
VT52 terminal, 3-17

W

Wildcard character
  file specification, 4-6 to 4-8

RSX-11M-PLUS
Command Language Manual
AA-FD04B-TC

READER'S COMMENTS

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

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.

_____

_____

_____

_____

_____

_____

What kind of user are you? ____ Programmer ____ Nonprogrammer

Years of experience as a computer programmer/user: _____

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

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here ----------

RSX-11M-PLUS
Command Language Manual
AA-FD04B-TC

READER'S COMMENTS

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

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.

_____

_____

_____

_____

_____

_____

What kind of user are you? ____ Programmer ____ Nonprogrammer

Years of experience as a computer programmer/user: _____

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

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

--- Do Not Tear - Fold Here ---